ELEX 7660
Project Report

Audio Visualizer

Eduardo Cabrera

Jacob Mbila

Digital Design Project
ELEX 7660
17th April 2017

AUDIO VISUALIZER

Introduction

One of the oldest forms of art is music. Music has defined many cultures, from old poems sang in the streets of Europe to Opera, Jazz, Soul, Rock, Hip Hop, Electronic and Pop, just to name a few genres, music is an art almost as old as civilization. The advancement in technology has played a big part in music, currently most of the music being played on the radio has been created electronically, music producers have devices that can compose a beat to which a vocal is added and hence music is made.

Sound is a quite simply defined as a vibration that is transferred as a mechanical wave through a medium (air). Sound can be simulated electronically as audio signal waveforms with precise amplitudes and frequencies.

We chose to do audio visualizer design project because of our deep infatuation with music. Music is a cultural phenomenon that has continued to reinvent itself time and time again, being able to visualize it and the process of making a device that provides visual aid brings a glimpse perspective on whole concept of music. Moreover, these devices usually have an aesthetic purpose; they are useful in many sound systems, they allow us to see the amplitude of a certain tone that we wish to attenuate or amplify. Audio visualizers are also used in audio devices such as mixers and equalizers.

Design process

An audio visualizer takes an input audio signal such as a song and displays a series of bars at different amplitudes and frequencies in accordance to the melody of the song. This project attempts to do just so using an RGB OLED 16-bit 96x64 display and a DE0 Nano FPGA board. Design for the audio visualizer includes five main parts: Input circuitry, A/D converter, FIFO, software and display.

INPUT CIRCUITRY:

For input into the audio visualizer music from a mobile audio player such as cellphone is the desired input audio signal. Just like many analog signals, an audio signal has positive and negative amplitudes. An A/D converter would only register the positive part of the audio signal and register the negative part as zero voltage. To overcome this inconvenience in the design a shifter circuit is added to shift the audio signal so that both parts of the signal can be registered by the A/D converter. Moreover, to get a decent amplitude on the LED display the design also incorporated an amplifier circuit to make sure the bars produced on the LED screen have a decent resolution.

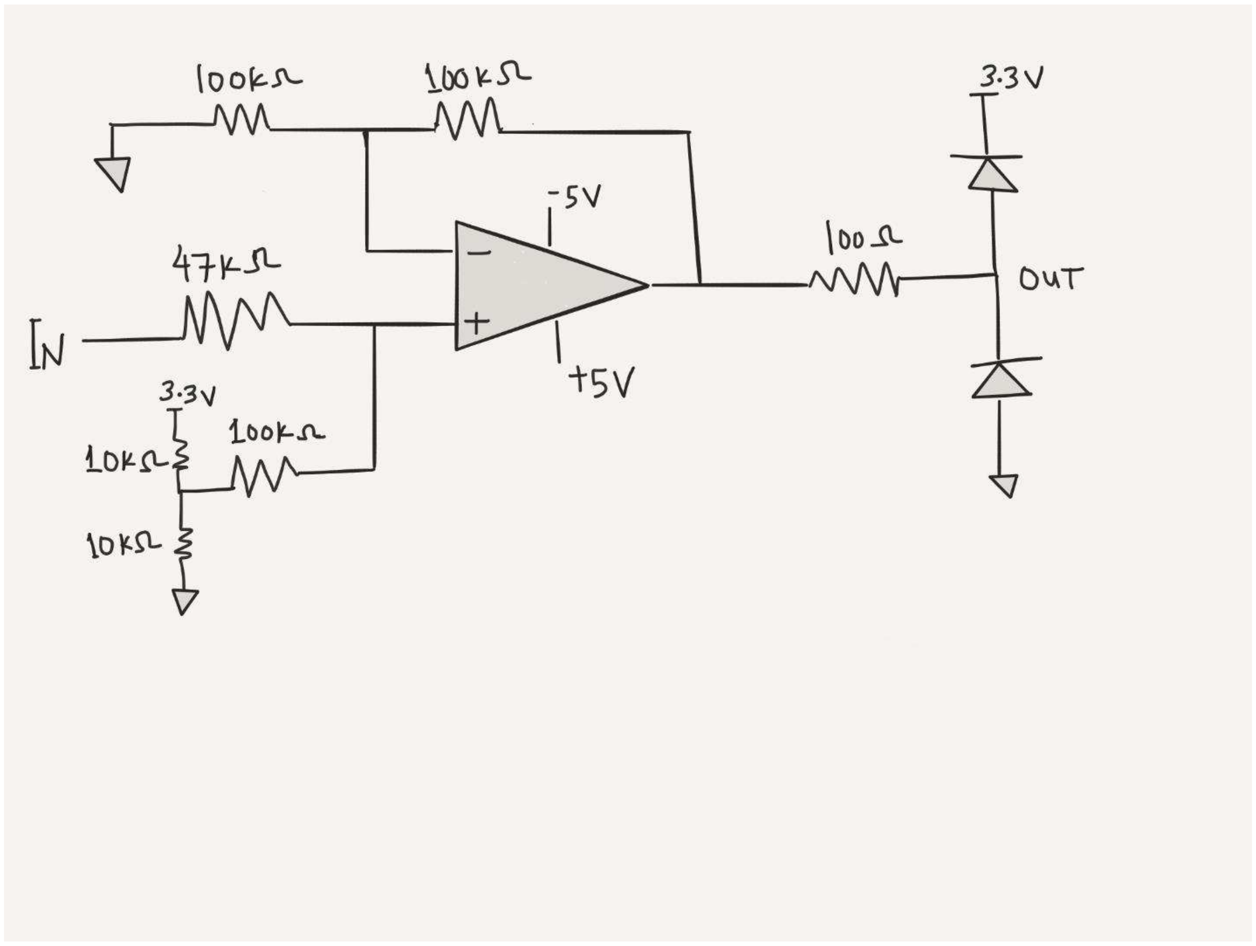


Diagram 1: Input Circuit Schematic

A/D CONVERTER:

For the Analog to Digital conversion part, we used the A/D Converter included in the FPGA board. It has a 12-bit resolution and 8 available analog channels. Sampling frequency of 50ksps to 200ksps. For this project, we used two analog inputs (IN0 & IN1) and a sampling rate of 98ksps which complies with the Nyquist sampling rate(40kHz). The connection for the conversion is located on the 2x13 header, the pins used are:

- VCC: PIN_1 (3.3V)
- GND: PIN_26
- IN_0: PIN_24
- IN_1: PIN_25

Samples are taken every 10.2 microseconds from one channel. Which means that both samples are taken and passed to the next stage every 20.4 microseconds. The sampling frequency is found by the following formula:

$$f_{sampling} = \frac{f_{clk}}{16*n*cnt}$$

where: n = # of inputs (2)
cnt = delay (16)

FIFO:

The FIFO (First-In-First-Out) acts as a RAM memory which organizes the rate of data transmission. It takes information at a certain rate and sends the information when the receiver is ready to receive, keeping record of which information was send first. For this project, we simulated a 20x16 bit RAM buffer. Each of the 16 RAM spaces represent a different sample taken by the ADC. The 20 bits represent data in the following way:

- LS 16-bits: The 16 less significant bits represent the discrete sample of the song.
- MS 4-bit: The 4 most significant bits represent the sample number.

These information is sent to the Avalon Bus to be modified by software. The key part of this part was the use of the extra 4 bits to transmit the sample number which was needed to time the connection between both phases.

SOFTWARE:

The software part is divided into two subsections: FFT & DISPLAY

FFT:

The FFT or Fast Fourier Transform converts a 2^n time samples to 2^n frequency samples. In this project, we take 16 time samples and convert it to 16 frequency samples. The algorithm of this operation is beyond the scope of this report. The code for the FFT can be found online and the source is listed on the references and accreditations go to the author: Emin Martinian. However, few modifications were added to suit our application. These changes are pointed out on the code.

Other sources can be found online or in the INTEL webpage. QUARTUS Prime also provides a IP module for the FFT where different parameters can be set.

DISPLAY:

The display used for this project is a RGB OLED 16-bit 96x64 display. The 16-bits are: 5-bit Red, 6-bit Green, 5-bit Blue. Each combination of these 3 sections produces a different color. The code for this section works by sending the value of each pixel to the display. To perform this function 4 key elements were used:

1. 2-D array: this array simulates the screen of the display. This array is set with the desired image and then set to the display.
2. Array of colors: this is a 16-value array which contains the color of each column.
3. Clean function: clears the array before setting the new image.
4. Fill_row function: fills the array with the bars that will be displayed in the display. It takes the level and number of the column and fills the array with each bar. The value of each bar is gotten from the FFT. The position and the colors are gotten from the sample number and the color array.

The array gotten at the end of this process is sent to the SPI to be sent to the display.

SPI:

The SPI is a Serial Peripheral Interface. It connects the FPGA board to a slave device. This module connects the software phase to the hardware phase using the Avalon Bus. This module oversees the connection between the board and the display. The information received by the SPI is gotten from the software part and then it is sent to the display when it is ready to receive.

The frequency of sound a human ear can comprehend ranges from 20 Hz to 20 kHz, an analog to digital converter with at least 40kHz sampling frequency is suitable for this project. DE0 – Nano FPGA board has an 8 channel 12-bit A/D converter with sampling rate 50ksps to 200ksps. Being as this A/D converter is well within the sampling frequency range 98 kHz is used with 2 channels as audio signal inputs

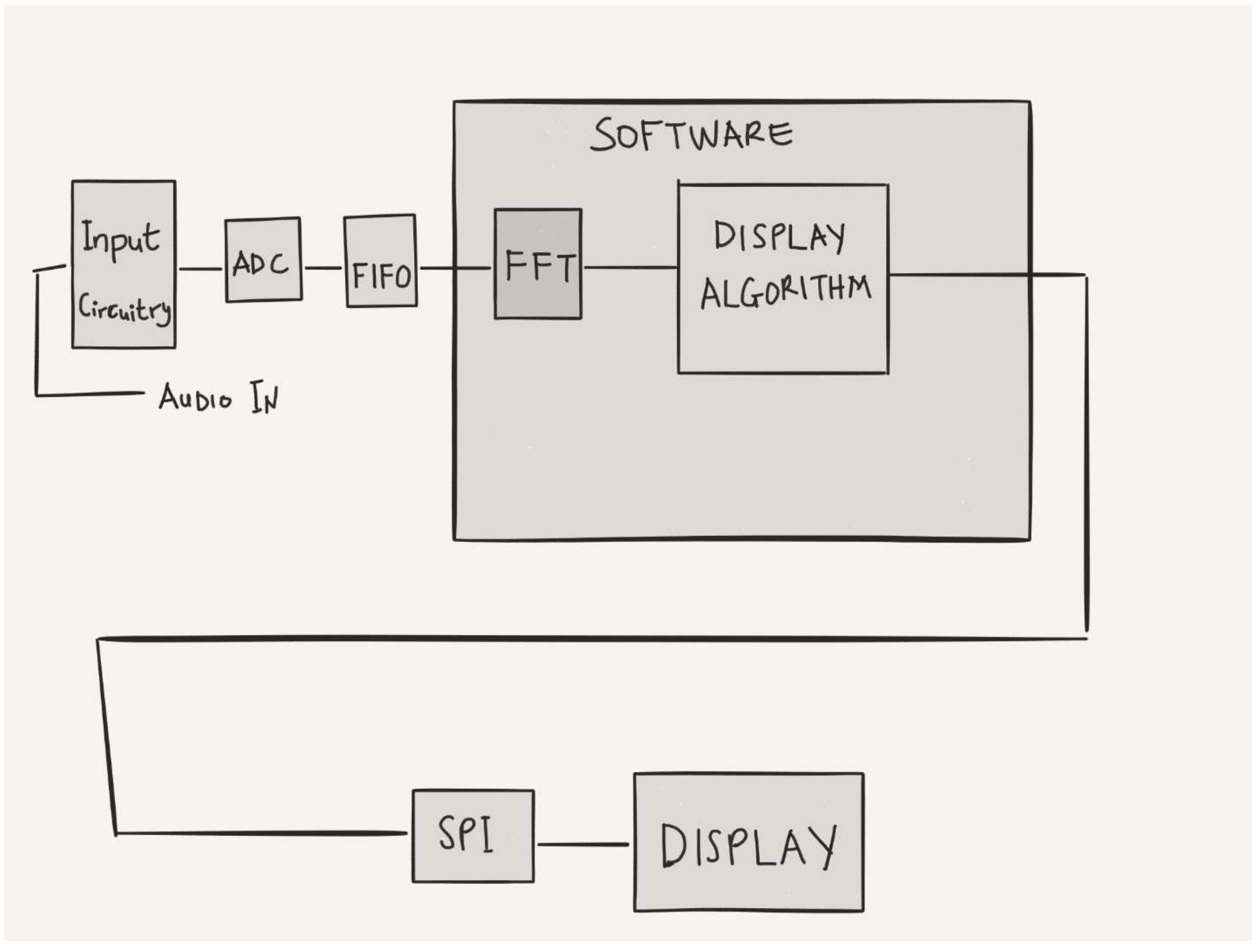
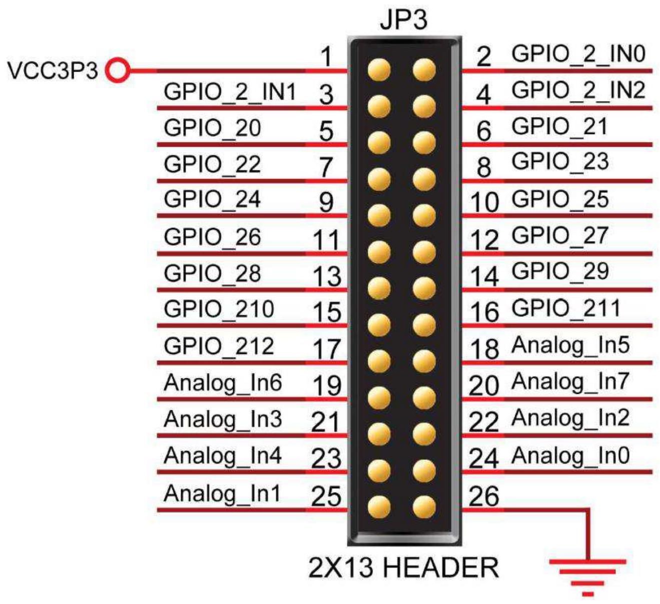


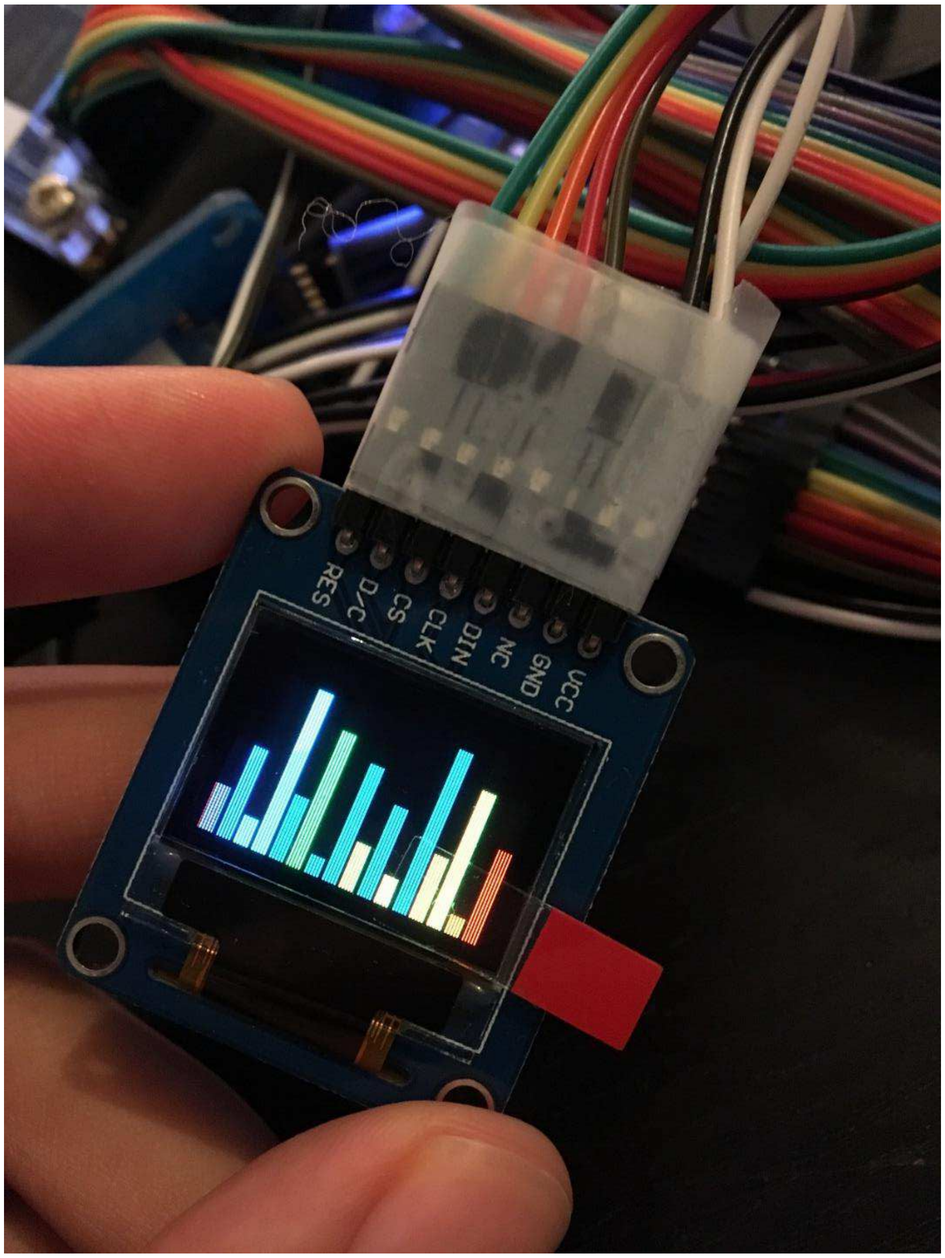
Diagram 2: Audio Visualizer Block Diagram

Appendix

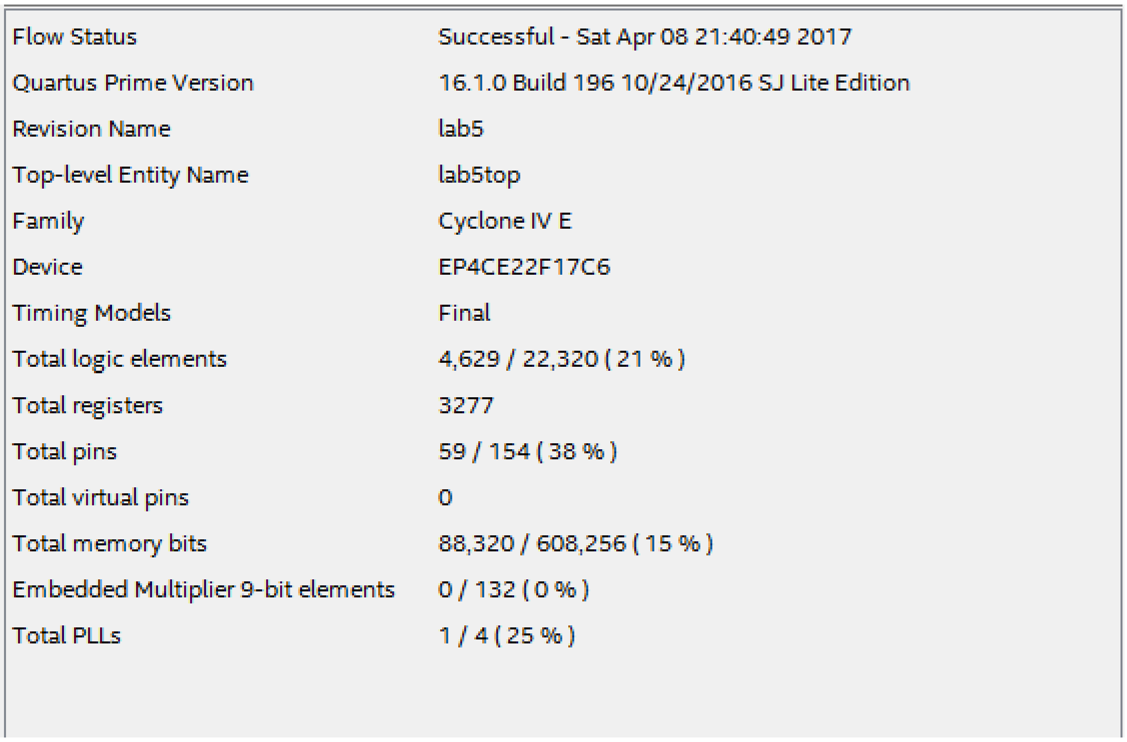
1. Pin distribution of the 2x13 header on the DE0 Nano FPGA board



2. Audio visualizer running

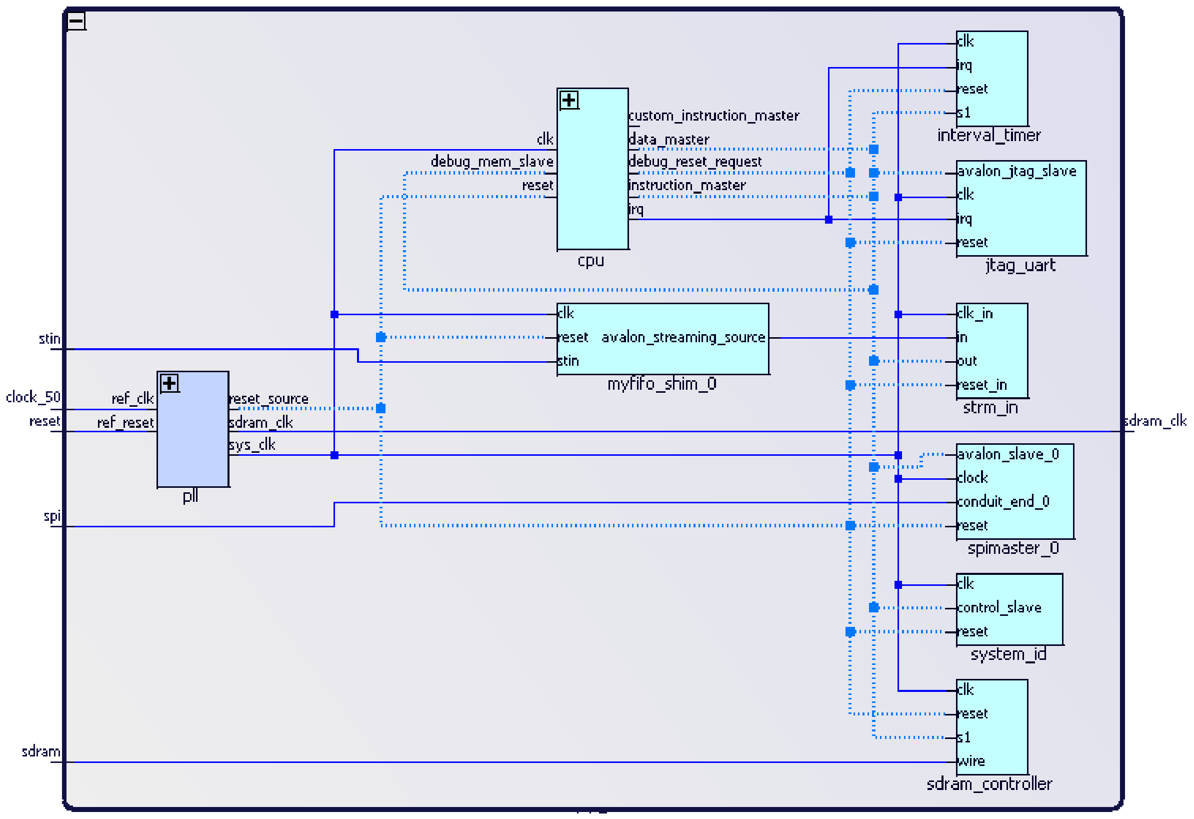


3. Compilation summary



Flow Status	Successful - Sat Apr 08 21:40:49 2017
Quartus Prime Version	16.1.0 Build 196 10/24/2016 SJ Lite Edition
Revision Name	lab5
Top-level Entity Name	lab5top
Family	Cyclone IV E
Device	EP4CE22F17C6
Timing Models	Final
Total logic elements	4,629 / 22,320 (21 %)
Total registers	3277
Total pins	59 / 154 (38 %)
Total virtual pins	0
Total memory bits	88,320 / 608,256 (15 %)
Embedded Multiplier 9-bit elements	0 / 132 (0 %)
Total PLLs	1 / 4 (25 %)

4. Block diagram



5. FIFO system verilog code

```
// This module is a FIFO interface for the FPGA. It has one 17x16 bit ram
buffer.

module myfifo
   (
     output logic iready,         // ready/valid input
     input  logic ivalid,
     input  logic [31:0] idata,

     input  logic oready,        // Avalon-ST output
     output logic ovalid,
     output logic [31:0] odata,

     input logic reset, clk ) ;


   reg [19:0] RAM [15:0];
   reg [3:0] readpointer;
   reg [3:0] writepointer;
   reg [3:0] readpointer_N;
   reg [3:0] writepointer_N;
   always_comb begin
      ovalid = (readpointer != writepointer);
      iready = (readpointer == 0) ? !(writepointer == 15) :
!((writepointer + 1) == readpointer);
      readpointer_N = !((oready == 1) && (ovalid == 1)) ? readpointer :
(readpointer == 15) ? 0 : readpointer + 1;
      writepointer_N = !((iready == 1) && (ivalid == 1)) ? writepointer :
(writepointer == 15) ? 0 : writepointer + 1;
   end

   always_ff @(posedge clk) begin

      if(reset) begin
            readpointer <= 0;
            writepointer <= 0;
      end
       RAM[0][19:16] = 4'd0;
       RAM[1][19:16] = 4'd1;
      RAM[2][19:16] = 4'd2;
      RAM[3][19:16] = 4'd3;
      RAM[4][19:16] = 4'd4;
      RAM[5][19:16] = 4'd5;
      RAM[6][19:16] = 4'd6;
       RAM[7][19:16] = 4'd7;
       RAM[8][19:16] = 4'd8;
       RAM[9][19:16] = 4'd9;
       RAM[10][19:16] = 4'd10;
      RAM[11][19:16] = 4'd11;
       RAM[12][19:16] = 4'd12;
      RAM[13][19:16] = 4'd13;
      RAM[14][19:16] = 4'd14;
      RAM[15][19:16] = 4'd15;

      RAM[writepointer] = idata[15:0];
      odata <= RAM[readpointer_N];

      if((iready == 1) && (ivalid == 1)) begin
            writepointer <= writepointer_N;
      end

      if((ovalid == 1) && (oready == 1)) begin
            readpointer <= readpointer_N;
      end

      end
endmodule
```

6. A/D Converter system verilog code

```
module lab5top
```

```
(
 input  logic CLOCK_50,
 output logic [7:0] LED,
 input  logic [1:0] KEY,

 // ADC SPI interface

 output logic ADC_CS_N,       // ssn
 output logic ADC_SADDR,      // mosi
 output logic ADC_SCLK,       // sclk
 input  logic ADC_SDAT,       // miso


 output logic [12:0] DRAM_ADDR,   //  dram.addr
 output logic [1:0]  DRAM_BA,     //      .ba
 output logic        DRAM_CAS_N,  //      .cas_n
 output logic        DRAM_CKE,    //      .cke
 output logic        DRAM_CS_N,   //      .cs_n
 inout  logic [15:0] DRAM_DQ,     //      .dq
 output logic [1:0]  DRAM_DQM,    //      .dqm
 output logic        DRAM_RAS_N,  //      .ras_n
 output logic        DRAM_WE_N,   //      .we_n
 output logic DRAM_CLK,
    output logic rgb_din,
    output logic rgb_clk, rgb_cs, rgb_dc, rgb_res

 ) ;

 // instantiates a Nios 2 processor with SDRAM memory and
 // ready/valid input for 'myfifo'. System defined in
 // lab5.qsys.

 lab5 u0
   (
    .clock_50_clk  (clk),     //  clock_50.clk
    .reset_reset   (reset_n), //  reset.reset

    .stin_idata(data),      // connected to myfifoshim stream inputs
    .stin_iready(ready),
    .stin_ivalid(valid),

    .sdram_clk_clk (DRAM_CLK), // sdram_clk.clk
    .sdram_addr    (DRAM_ADDR),    //     sdram.addr
    .sdram_ba      (DRAM_BA),      //          .ba
    .sdram_cas_n   (DRAM_CAS_N),   //          .cas_n
    .sdram_cke     (DRAM_CKE),     //          .cke
    .sdram_cs_n    (DRAM_CS_N),    //          .cs_n
    .sdram_dq      (DRAM_DQ),      //          .dq
    .sdram_dqm     (DRAM_DQM),     //          .dqm
    .sdram_ras_n   (DRAM_RAS_N),   //          .ras_n
    .sdram_we_n    (DRAM_WE_N),     //          .we_n
    .spi_sclk (rgb_clk),
          .spi_mosi (rgb_din),
          .spi_csn (rgb_cs),
          .spi_dcn (rgb_dc),
          .spi_resetn (rgb_res)
    );
   // Instatiate an SPI master for the DE0-Nano ADC. Reads from
   // ADC SPI pins, outputs to a ready/valid interface that feeds
      // 'myfifo'.

   logic ready ;
   logic valid ;
   logic [31:0] data ;
   logic reset_n, clk ;

   assign clk = CLOCK_50 ;
   assign reset_n = KEY[0] ;

   adcspi a0
     (
      .sclk(ADC_SCLK), .mosi(ADC_SADDR), .ssn(ADC_CS_N), // SPI master
      .miso(ADC_SDAT),

      .ready(ready),
      .valid(valid),          // data out
      .data(data),

      .clk(clk), .reset(~reset_n) ) ;

   // copy MS ADC bits to LEDs for debug
   assign LED = { data[27:24], data[11:8] } ;

endmodule

// -- start of adcspi.sv ---

// SPI master interface for TI ADC128S022
// for ELEX 7660 201710 Lab 5
// Ed.Casas 2017-2-16

// reads channels 0 and 1
// sclk is clk is divided by 16
// output is 16-bit samples from channels 0 and 1
// samples packed into 32 bits (ch 0 in MS byte)

// ADC128S0022 interface:
// 16 bit transfers

// mosi and cs* change on falling edge of sclk
// mosi bits 13:11 are (next) channel number

// miso sampled on rising edge of sclk
// miso data is on ls 12 bits of miso

// sample rate is sclk rate / 16
// sample rate must be 50 to 200 kHz
// sclk rate must be 800 kHz to 3.2 MHz
// e.g. 50 MHz / 32 = 1.5625 MHz sclk, ~98kHz sampling
// mosi timing relative to rising edge of sclk:
// setup is >10ns, hold  >10ns

// miso timing is relative to falling edge of sclk:
// access is <27ns, hold ~4ns

module adcspi
   (
    output logic sclk, mosi, ssn, // SPI master
    input logic miso,

    input logic ready,           // ready/valid data out
    output logic valid,
    output logic [31:0] data,

    input logic clk, reset ) ;

   parameter MISO = {5'b00001,27'b0} ;

   // clock/bit counter
   struct packed {
      logic wordcnt ;
      logic [3:0] bitcnt ;
      logic sclk ;
      logic [3:0] clkcnt ; } cnt, cnt_next ;

   logic [31:0] sr ;            // shift register

   logic rising, falling, done ;

   assign sclk = cnt.sclk ;

   // done all bits
   assign done = cnt ==? '{'1,'1,'1,'1} ;

   // clock/bit counter
   assign cnt_next = ( reset || done ) ? '0 : cnt+1'b1 ;
   always@(posedge clk)
     cnt <= cnt_next ;

   assign rising  =  cnt_next.sclk && ~cnt.sclk ;
   assign falling = ~cnt_next.sclk &&  cnt.sclk ;

   always@(posedge clk) begin

      if ( falling )            // shift mosi out
        mosi <= sr[31] ;

      if ( rising )             // shift miso in
        sr <= {sr[30:0],miso} ;

      if ( done ) begin
         data <= sr ;           // copy to parallel out
         sr <= MISO ;           // channel select serial out
         mosi <= MISO[31] ;
         valid <= '1 ;          // data ready
      end
      if ( ready && valid )      // data was read
        valid <= '0 ;

   end

   always@(posedge clk)          // run continously
     ssn <= reset ;

endmodule
```

7. SPI master system verilog code

```
// spimaster.sv - SPI master to drive OLED display
// Ed.Casas 2017-1-30

// Modified by: Eduardo Cabrera

module spimaster
  #( N=16 )
   ( input logic [31:0] writedata, // Avalon MM bus
     output logic [31:0] readdata,
     input logic read, write,

     output logic sclk, mosi, csn, dcn, resetn, // SPI master

     input logic reset, clk ) ;

   logic [31:0] data, data_next ;      // data register
   logic [4:0] i, i_next, j, j_next ; // bit/delay counters

   // output FF inputs
   logic sclk_next, mosi_next, csn_next, dcn_next ;

   // registers
   always_ff@(posedge clk) begin
      data <= data_next ;        // data register

      sclk <= sclk_next ;        // SPI port

        csn <= csn_next;
        dcn <= dcn_next;
      mosi <= mosi_next;
      i <= i_next ;              // bit/clock counts
      j <= j_next ;

   end

   // connect master reset to SPI reset
   assign resetn = !reset ;
   // status register: idle is LS bit
   assign readdata = {31'b0,csn} ;

   // combinational logic
   always_comb begin
      data_next = write ? writedata : data ;

      // i=7...0
      i_next = write ? 5'd7 : ( j==N/2-1 && sclk ) ? i-1 : i ;


      // sclk=0..1
      sclk_next = write ? '0 : ( j==N/2-1 ) ? ~sclk : sclk ;
      // j=0...N/2-1
      j_next = write ? '0 : ( j==N/2-1 ) ? '0 : j+1 ;



      mosi_next = write ? data[i] : ( j==N/2-1 && sclk) ? data[i_next] :
      data[i];
      csn_next = !resetn ? 1 : write ? '0 : ( !i && j==N/2-1 && sclk ) ?
      5'd1 : csn ;
      dcn_next = ~data_next[8];
   end

endmodule
```

8. Pin assignment file

```
set_global_assignment -name FAMILY "Cyclone IV E"
set_global_assignment -name DEVICE EP4CE22F17C6
set_global_assignment -name TOP_LEVEL_ENTITY lab5top
set_global_assignment -name ORIGINAL_QUARTUS_VERSION 16.1.0
set_global_assignment -name PROJECT_CREATION_TIME_DATE "00:42:31  FEBRUARY
14, 2017"
set_global_assignment -name LAST_QUARTUS_VERSION "16.1.0 Lite Edition"
set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output_files
set_global_assignment -name MIN_CORE_JUNCTION_TEMP 0
set_global_assignment -name MAX_CORE_JUNCTION_TEMP 85
set_location_assignment PIN_R8 -to CLOCK_50
set_location_assignment PIN_A15 -to LED[0]
set_location_assignment PIN_A13 -to LED[1]
set_location_assignment PIN_B13 -to LED[2]
set_location_assignment PIN_A11 -to LED[3]
set_location_assignment PIN_D1 -to LED[4]
set_location_assignment PIN_F3 -to LED[5]
set_location_assignment PIN_B1 -to LED[6]
set_location_assignment PIN_L3 -to LED[7]
set_location_assignment PIN_J15 -to KEY[0]
set_location_assignment PIN_E1 -to KEY[1]
set_location_assignment PIN_P2 -to DRAM_ADDR[0]
set_location_assignment PIN_N5 -to DRAM_ADDR[1]
set_location_assignment PIN_N6 -to DRAM_ADDR[2]
set_location_assignment PIN_M8 -to DRAM_ADDR[3]
set_location_assignment PIN_P8 -to DRAM_ADDR[4]
set_location_assignment PIN_T7 -to DRAM_ADDR[5]
set_location_assignment PIN_N8 -to DRAM_ADDR[6]
set_location_assignment PIN_T6 -to DRAM_ADDR[7]
set_location_assignment PIN_R1 -to DRAM_ADDR[8]
set_location_assignment PIN_P1 -to DRAM_ADDR[9]
set_location_assignment PIN_N2 -to DRAM_ADDR[10]
set_location_assignment PIN_N1 -to DRAM_ADDR[11]
set_location_assignment PIN_L4 -to DRAM_ADDR[12]
set_location_assignment PIN_M7 -to DRAM_BA[0]
set_location_assignment PIN_M6 -to DRAM_BA[1]
set_location_assignment PIN_L7 -to DRAM_CKE
set_location_assignment PIN_R4 -to DRAM_CLK
set_location_assignment PIN_P6 -to DRAM_CS_N
set_location_assignment PIN_G2 -to DRAM_DQ[0]
set_location_assignment PIN_G1 -to DRAM_DQ[1]
set_location_assignment PIN_L8 -to DRAM_DQ[2]
set_location_assignment PIN_K5 -to DRAM_DQ[3]
set_location_assignment PIN_K2 -to DRAM_DQ[4]
set_location_assignment PIN_J2 -to DRAM_DQ[5]
set_location_assignment PIN_J1 -to DRAM_DQ[6]
set_location_assignment PIN_R7 -to DRAM_DQ[7]
set_location_assignment PIN_T4 -to DRAM_DQ[8]
set_location_assignment PIN_T2 -to DRAM_DQ[9]
set_location_assignment PIN_T3 -to DRAM_DQ[10]
set_location_assignment PIN_R3 -to DRAM_DQ[11]
set_location_assignment PIN_R5 -to DRAM_DQ[12]
set_location_assignment PIN_P3 -to DRAM_DQ[13]
set_location_assignment PIN_N3 -to DRAM_DQ[14]
set_location_assignment PIN_K1 -to DRAM_DQ[15]
set_location_assignment PIN_R6 -to DRAM_DQM[0]
set_location_assignment PIN_T5 -to DRAM_DQM[1]
set_location_assignment PIN_L1 -to DRAM_CAS_N
set_location_assignment PIN_L2 -to DRAM_RAS_N
set_location_assignment PIN_C2 -to DRAM_WE_N
set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id Top
set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL
PLACEMENT_AND_ROUTING -section_id Top
set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top
set_global_assignment -name POWER_PRESET_COOLING_SOLUTION "23 MM HEAT SINK
WITH 200 LFPM AIRFLOW"
set_global_assignment -name POWER_BOARD_THERMAL_MODEL "NONE (CONSERVATIVE)"
set_location_assignment PIN_A10 -to ADC_CS_N
set_location_assignment PIN_B10 -to ADC_SADDR
set_location_assignment PIN_B14 -to ADC_SCLK
set_location_assignment PIN_A9 -to ADC_SDAT


set_location_assignment PIN_D9 -to rgb_din
set_location_assignment PIN_E10 -to rgb_clk
set_location_assignment PIN_B11 -to rgb_cs
set_location_assignment PIN_D11 -to rgb_dc
set_location_assignment PIN_B12 -to rgb_res
```

```
set_global_assignment -name ENABLE_SIGNALTAP ON
set_global_assignment -name USE_SIGNALTAP_FILE lab5.stp
set_global_assignment -name SLD_NODE_CREATOR_ID 110 -section_id
auto_signaltap_0
set_global_assignment -name SLD_NODE_ENTITY_NAME sld_signaltap -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_clk -to
CLOCK_50 -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_RAM_BLOCK_TYPE=AUTO" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_NODE_INFO=805334528" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_POWER_UP_TRIGGER=0" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_STORAGE_QUALIFIER_INVERSION_MASK_LENGTH=0" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_SEGMENT_SIZE=1024" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_ATTRIBUTE_MEM_MODE=OFF" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_STATE_FLOW_USE_GENERATED=0" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT "SLD_STATE_BITS=11"
-section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_BUFFER_FULL_STOP=1" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_CURRENT_RESOURCE_WIDTH=1" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_INCREMENTAL_ROUTING=1" -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT crc[0]
-to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT crc[3]
-to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT crc[4]
-to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT crc[7]
-to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT crc[8]
-to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[10] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[13] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[14] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[16] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[18] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[19] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[23] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
```

```
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[24] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[25] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[28] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[29] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[30] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[31] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_TRIGGER_LEVEL=1" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_SAMPLE_DEPTH=1024" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_TRIGGER_IN_ENABLED=0" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_TRIGGER_PIPELINE=0" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_RAM_PIPELINE=0" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_COUNTER_PIPELINE=0" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_ADVANCED_TRIGGER_ENTITY=basic,1," -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_TRIGGER_LEVEL_PIPELINE=1" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_ENABLE_ADVANCED_TRIGGER=0" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT "SLD_DATA_BITS=70"
-section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_TRIGGER_BITS=70" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_STORAGE_QUALIFIER_BITS=70" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_INVERSION_MASK=000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000" -section_id auto_signaltap_0
set_global_assignment -name SLD_NODE_PARAMETER_ASSIGNMENT
"SLD_INVERSION_MASK_LENGTH=234" -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT crc[2]
-to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT crc[9]
-to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[11] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[17] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[20] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[21] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[22] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
```

```
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[26] -to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[27] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[0] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[0]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[1] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[10]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[2] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[11]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[3] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[12]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[4] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[13]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[5] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[14]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[6] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[15]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[7] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[16]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[8] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[17]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[9] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[18]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[10] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[19]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[11] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[1]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[12] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[20]" -section_id
auto_signaltap_0
```

```
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[13] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[21]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[14] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[22]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[15] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[23]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[16] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[24]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[17] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[25]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[18] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[26]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[19] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[27]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[20] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[28]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[21] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[29]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[22] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[2]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[23] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[30]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[24] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[31]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[25] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[3]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[26] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[4]" -section_id
auto_signaltap_0
```

```
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[27] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[5]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[28] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[6]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[29] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[7]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[30] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[8]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[31] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[9]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[32] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[0]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[33] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[10]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[34] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[11]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[35] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[12]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[36] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[13]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[37] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[14]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[38] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[15]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[39] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[16]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[40] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[17]" -section_id
auto_signaltap_0
```

```
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[41] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[18]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[42] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[19]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[43] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[1]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[44] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[20]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[45] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[21]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[46] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[22]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[47] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[23]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[48] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[24]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[49] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[25]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[50] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[26]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[51] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[27]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[52] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[28]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[53] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[29]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[54] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[2]" -section_id
auto_signaltap_0
```

```
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[55] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[30]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[56] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[31]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[57] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[3]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[58] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[4]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[59] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[5]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[60] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[6]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[61] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[7]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[62] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[8]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[63] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[9]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[64] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|readp[0]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[65] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|readp[1]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[66] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|readp[2]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[67] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|writep[0]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[68] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|writep[1]" -section_id
auto_signaltap_0
```

```
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT
acq_trigger_in[69] -to
"lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|writep[2]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[0]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[0]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[1]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[10]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[2]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[11]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[3]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[12]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[4]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[13]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[5]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[14]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[6]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[15]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[7]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[16]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[8]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[17]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[9]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[18]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[10]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[19]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[11]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[1]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[12]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[20]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[13]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[21]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[14]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[22]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[15]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[23]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[16]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[24]" -section_id
auto_signaltap_0
```

```
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[17]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[25]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[18]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[26]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[19]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[27]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[20]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[28]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[21]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[29]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[22]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[2]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[23]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[30]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[24]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[31]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[25]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[3]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[26]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[4]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[27]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[5]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[28]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[6]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[29]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[7]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[30]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[8]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[31]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|idata[9]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[32]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[0]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[33]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[10]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[34]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[11]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[35]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[12]" -section_id
auto_signaltap_0
```

```
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[36]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[13]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[37]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[14]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[38]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[15]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[39]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[16]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[40]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[17]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[41]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[18]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[42]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[19]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[43]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[1]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[44]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[20]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[45]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[21]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[46]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[22]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[47]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[23]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[48]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[24]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[49]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[25]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[50]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[26]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[51]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[27]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[52]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[28]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[53]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[29]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[54]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[2]" -section_id
auto_signaltap_0
```

```
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[55]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[30]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[56]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[31]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[57]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[3]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[58]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[4]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[59]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[5]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[60]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[6]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[61]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[7]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[62]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[8]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[63]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|odata[9]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[64]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|readp[0]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[65]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|readp[1]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[66]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|readp[2]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[67]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|writep[0]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[68]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|writep[1]" -section_id
auto_signaltap_0
set_instance_assignment -name CONNECT_TO_SLD_NODE_ENTITY_PORT acq_data_in[69]
-to "lab5:u0|myfifoshim:myfifo_shim_0|myfifo:fifo0|writep[2]" -section_id
auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT crc[1]
-to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT crc[5]
-to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT crc[6]
-to auto_signaltap_0|vcc -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[12] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name POST_FIT_CONNECT_TO_SLD_NODE_ENTITY_PORT
crc[15] -to auto_signaltap_0|gnd -section_id auto_signaltap_0
set_instance_assignment -name PARTITION_HIERARCHY root_partition -to | -
section_id Top
```

```
set_global_assignment -name SLD_FILE db/lab5_auto_stripped.stp
set_global_assignment -name SYSTEMVERILOG_FILE ../myfifo.sv
set_global_assignment -name VERILOG_FILE myfifoshim.v
set_global_assignment -name QIP_FILE lab5/synthesis/lab5.qip
set_global_assignment -name SYSTEMVERILOG_FILE lab5top.sv
set_global_assignment -name SDC_FILE lab5.sdc
set_global_assignment -name CDF_FILE lab5.cdf
set_global_assignment -name SIGNALTAP_FILE lab5.stp
```

9. Project file

```
// Generated using ACDS version 16.1 196

`timescale 1 ps / 1 ps
module lab5 (
		input  wire        clock_50_clk,  //  clock_50.clk
		input  wire        reset_reset,   //     reset.reset
		output wire [12:0] sdram_addr,    //     sdram.addr
		output wire [1:0]  sdram_ba,      //          .ba
		output wire        sdram_cas_n,   //          .cas_n
		output wire        sdram_cke,     //          .cke
		output wire        sdram_cs_n,    //          .cs_n
		inout  wire [15:0] sdram_dq,      //          .dq
		output wire [1:0]  sdram_dqm,     //          .dqm
		output wire        sdram_ras_n,   //          .ras_n
		output wire        sdram_we_n,    //          .we_n
		output wire        sdram_clk_clk, // sdram_clk.clk
		output wire        spi_sclk,      //       spi.sclk
		output wire        spi_mosi,      //          .mosi
		output wire        spi_csn,       //          .csn
		output wire        spi_dcn,       //          .dcn
		output wire        spi_resetn,    //          .resetn
		output wire        stin_iready,   //      stin.iready
		input  wire        stin_ivalid,   //          .ivalid
		input  wire [31:0] stin_idata     //          .idata
	);

	wire         pll_sys_clk_clk;
// pll:sys_clk_clk -> [avalon_st_adapter:in_clk_0_clk, cpu:clk,
interval_timer:clk, irq_mapper:clk, jtag_uart:clk,
mm_interconnect_0:pll_sys_clk_clk, myfifo_shim_0:clk, rst_controller:clk,
rst_controller_001:clk, sdram_controller:clk, spimaster_0:clk,
strm_in:wrclock, system_id:clock]
	wire  [31:0] cpu_data_master_readdata;
// mm_interconnect_0:cpu_data_master_readdata -> cpu:d_readdata
	wire         cpu_data_master_waitrequest;
// mm_interconnect_0:cpu_data_master_waitrequest -> cpu:d_waitrequest
	wire         cpu_data_master_debugaccess;
// cpu:debug_mem_slave_debugaccess_to_roms ->
mm_interconnect_0:cpu_data_master_debugaccess
	wire  [28:0] cpu_data_master_address;
// cpu:d_address -> mm_interconnect_0:cpu_data_master_address
	wire   [3:0] cpu_data_master_byteenable;
// cpu:d_byteenable -> mm_interconnect_0:cpu_data_master_byteenable
```

```
      wire         cpu_data_master_read;
// cpu:d_read -> mm_interconnect_0:cpu_data_master_read
      wire         cpu_data_master_write;
// cpu:d_write -> mm_interconnect_0:cpu_data_master_write
      wire  [31:0] cpu_data_master_writedata;
// cpu:d_writedata -> mm_interconnect_0:cpu_data_master_writedata
      wire  [31:0] cpu_instruction_master_readdata;
// mm_interconnect_0:cpu_instruction_master_readdata -> cpu:i_readdata
      wire         cpu_instruction_master_waitrequest;
// mm_interconnect_0:cpu_instruction_master_waitrequest -> cpu:i_waitrequest
      wire  [26:0] cpu_instruction_master_address;
// cpu:i_address -> mm_interconnect_0:cpu_instruction_master_address
      wire         cpu_instruction_master_read;
// cpu:i_read -> mm_interconnect_0:cpu_instruction_master_read
      wire         mm_interconnect_0_jtag_uart_avalon_jtag_slave_chipselect;
// mm_interconnect_0:jtag_uart_avalon_jtag_slave_chipselect ->
jtag_uart:av_chipselect
      wire  [31:0] mm_interconnect_0_jtag_uart_avalon_jtag_slave_readdata;
// jtag_uart:av_readdata ->
mm_interconnect_0:jtag_uart_avalon_jtag_slave_readdata
      wire         mm_interconnect_0_jtag_uart_avalon_jtag_slave_waitrequest;
// jtag_uart:av_waitrequest ->
mm_interconnect_0:jtag_uart_avalon_jtag_slave_waitrequest
      wire   [0:0] mm_interconnect_0_jtag_uart_avalon_jtag_slave_address;
// mm_interconnect_0:jtag_uart_avalon_jtag_slave_address ->
jtag_uart:av_address
      wire         mm_interconnect_0_jtag_uart_avalon_jtag_slave_read;
// mm_interconnect_0:jtag_uart_avalon_jtag_slave_read -> jtag_uart:av_read_n
      wire         mm_interconnect_0_jtag_uart_avalon_jtag_slave_write;
// mm_interconnect_0:jtag_uart_avalon_jtag_slave_write ->
jtag_uart:av_write_n
      wire  [31:0] mm_interconnect_0_jtag_uart_avalon_jtag_slave_writedata;
// mm_interconnect_0:jtag_uart_avalon_jtag_slave_writedata ->
jtag_uart:av_writedata
      wire  [31:0] mm_interconnect_0_spimaster_0_avalon_slave_0_readdata;
// spimaster_0:avs_readdata ->
mm_interconnect_0:spimaster_0_avalon_slave_0_readdata
      wire         mm_interconnect_0_spimaster_0_avalon_slave_0_read;
// mm_interconnect_0:spimaster_0_avalon_slave_0_read -> spimaster_0:avs_read
      wire         mm_interconnect_0_spimaster_0_avalon_slave_0_write;
// mm_interconnect_0:spimaster_0_avalon_slave_0_write ->
spimaster_0:avs_write
      wire  [31:0] mm_interconnect_0_spimaster_0_avalon_slave_0_writedata;
// mm_interconnect_0:spimaster_0_avalon_slave_0_writedata ->
spimaster_0:avs_writedata
      wire  [31:0] mm_interconnect_0_system_id_control_slave_readdata;
// system_id:readdata -> mm_interconnect_0:system_id_control_slave_readdata
      wire   [0:0] mm_interconnect_0_system_id_control_slave_address;
// mm_interconnect_0:system_id_control_slave_address -> system_id:address
      wire  [31:0] mm_interconnect_0_cpu_debug_mem_slave_readdata;
// cpu:debug_mem_slave_readdata ->
mm_interconnect_0:cpu_debug_mem_slave_readdata
      wire         mm_interconnect_0_cpu_debug_mem_slave_waitrequest;
// cpu:debug_mem_slave_waitrequest ->
mm_interconnect_0:cpu_debug_mem_slave_waitrequest
```

```
      wire         mm_interconnect_0_cpu_debug_mem_slave_debugaccess;
// mm_interconnect_0:cpu_debug_mem_slave_debugaccess ->
cpu:debug_mem_slave_debugaccess
      wire   [8:0] mm_interconnect_0_cpu_debug_mem_slave_address;
// mm_interconnect_0:cpu_debug_mem_slave_address ->
cpu:debug_mem_slave_address
      wire         mm_interconnect_0_cpu_debug_mem_slave_read;
// mm_interconnect_0:cpu_debug_mem_slave_read -> cpu:debug_mem_slave_read
      wire   [3:0] mm_interconnect_0_cpu_debug_mem_slave_byteenable;
// mm_interconnect_0:cpu_debug_mem_slave_byteenable ->
cpu:debug_mem_slave_byteenable
      wire         mm_interconnect_0_cpu_debug_mem_slave_write;
// mm_interconnect_0:cpu_debug_mem_slave_write -> cpu:debug_mem_slave_write
      wire  [31:0] mm_interconnect_0_cpu_debug_mem_slave_writedata;
// mm_interconnect_0:cpu_debug_mem_slave_writedata ->
cpu:debug_mem_slave_writedata
      wire  [31:0] mm_interconnect_0_strm_in_out_readdata;
// strm_in:avalonmm_read_slave_readdata ->
mm_interconnect_0:strm_in_out_readdata
      wire         mm_interconnect_0_strm_in_out_waitrequest;
// strm_in:avalonmm_read_slave_waitrequest ->
mm_interconnect_0:strm_in_out_waitrequest
      wire   [0:0] mm_interconnect_0_strm_in_out_address;
// mm_interconnect_0:strm_in_out_address ->
strm_in:avalonmm_read_slave_address
      wire         mm_interconnect_0_strm_in_out_read;
// mm_interconnect_0:strm_in_out_read -> strm_in:avalonmm_read_slave_read
      wire         mm_interconnect_0_interval_timer_s1_chipselect;
// mm_interconnect_0:interval_timer_s1_chipselect ->
interval_timer:chipselect
      wire  [15:0] mm_interconnect_0_interval_timer_s1_readdata;
// interval_timer:readdata -> mm_interconnect_0:interval_timer_s1_readdata
      wire   [2:0] mm_interconnect_0_interval_timer_s1_address;
// mm_interconnect_0:interval_timer_s1_address -> interval_timer:address
      wire         mm_interconnect_0_interval_timer_s1_write;
// mm_interconnect_0:interval_timer_s1_write -> interval_timer:write_n
      wire  [15:0] mm_interconnect_0_interval_timer_s1_writedata;
// mm_interconnect_0:interval_timer_s1_writedata -> interval_timer:writedata
      wire         mm_interconnect_0_sdram_controller_s1_chipselect;
// mm_interconnect_0:sdram_controller_s1_chipselect -> sdram_controller:az_cs
      wire  [15:0] mm_interconnect_0_sdram_controller_s1_readdata;
// sdram_controller:za_data -> mm_interconnect_0:sdram_controller_s1_readdata
      wire         mm_interconnect_0_sdram_controller_s1_waitrequest;
// sdram_controller:za_waitrequest ->
mm_interconnect_0:sdram_controller_s1_waitrequest
      wire  [23:0] mm_interconnect_0_sdram_controller_s1_address;
// mm_interconnect_0:sdram_controller_s1_address -> sdram_controller:az_addr
      wire         mm_interconnect_0_sdram_controller_s1_read;
// mm_interconnect_0:sdram_controller_s1_read -> sdram_controller:az_rd_n
      wire   [1:0] mm_interconnect_0_sdram_controller_s1_byteenable;
// mm_interconnect_0:sdram_controller_s1_byteenable ->
sdram_controller:az_be_n
      wire         mm_interconnect_0_sdram_controller_s1_readdatavalid;
// sdram_controller:za_valid ->
mm_interconnect_0:sdram_controller_s1_readdatavalid
      wire         mm_interconnect_0_sdram_controller_s1_write;
// mm_interconnect_0:sdram_controller_s1_write -> sdram_controller:az_wr_n
```

```
      wire  [15:0] mm_interconnect_0_sdram_controller_s1_writedata;
// mm_interconnect_0:sdram_controller_s1_writedata ->
sdram_controller:az_data
      wire         irq_mapper_receiver0_irq;
// interval_timer:irq -> irq_mapper:receiver0_irq
      wire         irq_mapper_receiver1_irq;
// jtag_uart:av_irq -> irq_mapper:receiver1_irq
      wire  [31:0] cpu_irq_irq;
// irq_mapper:sender_irq -> cpu:irq
      wire         myfifo_shim_0_avalon_streaming_source_valid;
// myfifo_shim_0:valid -> avalon_st_adapter:in_0_valid
      wire  [31:0] myfifo_shim_0_avalon_streaming_source_data;
// myfifo_shim_0:data -> avalon_st_adapter:in_0_data
      wire         myfifo_shim_0_avalon_streaming_source_ready;
// avalon_st_adapter:in_0_ready -> myfifo_shim_0:ready
      wire         avalon_st_adapter_out_0_valid;
// avalon_st_adapter:out_0_valid -> strm_in:avalonst_sink_valid
      wire  [31:0] avalon_st_adapter_out_0_data;
// avalon_st_adapter:out_0_data -> strm_in:avalonst_sink_data
      wire         avalon_st_adapter_out_0_ready;
// strm_in:avalonst_sink_ready -> avalon_st_adapter:out_0_ready
      wire         rst_controller_reset_out_reset;
// rst_controller:reset_out -> [cpu:reset_n, interval_timer:reset_n,
irq_mapper:reset, jtag_uart:rst_n,
mm_interconnect_0:cpu_reset_reset_bridge_in_reset_reset, system_id:reset_n]
      wire         cpu_debug_reset_request_reset;
// cpu:debug_reset_request -> rst_controller:reset_in0
      wire         pll_reset_source_reset;
// pll:reset_source_reset -> [rst_controller:reset_in1,
rst_controller_001:reset_in0]
      wire         rst_controller_001_reset_out_reset;
// rst_controller_001:reset_out -> [avalon_st_adapter:in_rst_0_reset,
mm_interconnect_0:spimaster_0_reset_reset_bridge_in_reset_reset,
myfifo_shim_0:reset, sdram_controller:reset_n, spimaster_0:reset,
strm_in:reset_n]

      lab5_cpu cpu (
            .clk                                 (pll_sys_clk_clk),
//                         clk.clk
            .reset_n
(~rst_controller_reset_out_reset),                       //
reset.reset_n
            .d_address                           (cpu_data_master_address),
//                 data_master.address
            .d_byteenable
(cpu_data_master_byteenable),                            //
.byteenable
            .d_read                              (cpu_data_master_read),
//                            .read
            .d_readdata                          (cpu_data_master_readdata),
//                            .readdata
            .d_waitrequest
(cpu_data_master_waitrequest),                           //
.waitrequest
            .d_write                             (cpu_data_master_write),
//                            .write
```

```
            .d_writedata                         (cpu_data_master_writedata),
//                          .writedata
            .debug_mem_slave_debugaccess_to_roms
(cpu_data_master_debugaccess),                          //
.debugaccess
            .i_address
(cpu_instruction_master_address),                       //
instruction_master.address
            .i_read
(cpu_instruction_master_read),                          //
.read
            .i_readdata
(cpu_instruction_master_readdata),                      //
.readdata
            .i_waitrequest
(cpu_instruction_master_waitrequest),                   //
.waitrequest
            .irq                                 (cpu_irq_irq),
//                       irq.irq
            .debug_reset_request
(cpu_debug_reset_request_reset),                        //
debug_reset_request.reset
            .debug_mem_slave_address
(mm_interconnect_0_cpu_debug_mem_slave_address),        //
debug_mem_slave.address
            .debug_mem_slave_byteenable
(mm_interconnect_0_cpu_debug_mem_slave_byteenable),     //
.byteenable
            .debug_mem_slave_debugaccess
(mm_interconnect_0_cpu_debug_mem_slave_debugaccess),    //
.debugaccess
            .debug_mem_slave_read
(mm_interconnect_0_cpu_debug_mem_slave_read),           //
.read
            .debug_mem_slave_readdata
(mm_interconnect_0_cpu_debug_mem_slave_readdata),       //
.readdata
            .debug_mem_slave_waitrequest
(mm_interconnect_0_cpu_debug_mem_slave_waitrequest),    //
.waitrequest
            .debug_mem_slave_write
(mm_interconnect_0_cpu_debug_mem_slave_write),          //
.write
            .debug_mem_slave_writedata
(mm_interconnect_0_cpu_debug_mem_slave_writedata),      //
.writedata
            .dummy_ci_port                       ()
// custom_instruction_master.readra
      );

      lab5_interval_timer interval_timer (
            .clk        (pll_sys_clk_clk),                              //
clk.clk
            .reset_n    (~rst_controller_reset_out_reset),              //
reset.reset_n
            .address    (mm_interconnect_0_interval_timer_s1_address),    //
s1.address
```

```
            .writedata  (mm_interconnect_0_interval_timer_s1_writedata),  //
.writedata
            .readdata   (mm_interconnect_0_interval_timer_s1_readdata),   //
.readdata
            .chipselect (mm_interconnect_0_interval_timer_s1_chipselect), //
.chipselect
            .write_n    (~mm_interconnect_0_interval_timer_s1_write),     //
.write_n
            .irq        (irq_mapper_receiver0_irq)                        //
irq.irq
      );

      lab5_jtag_uart jtag_uart (
            .clk            (pll_sys_clk_clk),
//                 clk.clk
            .rst_n          (~rst_controller_reset_out_reset),
//               reset.reset_n
            .av_chipselect
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_chipselect),  //
avalon_jtag_slave.chipselect
            .av_address
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_address),     //
.address
            .av_read_n
(~mm_interconnect_0_jtag_uart_avalon_jtag_slave_read),       //
.read_n
            .av_readdata
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_readdata),    //
.readdata
            .av_write_n
(~mm_interconnect_0_jtag_uart_avalon_jtag_slave_write),      //
.write_n
            .av_writedata
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_writedata),   //
.writedata
            .av_waitrequest
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_waitrequest), //
.waitrequest
            .av_irq         (irq_mapper_receiver1_irq)
//                 irq.irq
      );

      myfifoshim myfifo_shim_0 (
            .reset  (rst_controller_001_reset_out_reset),          //
reset.reset
            .ready  (myfifo_shim_0_avalon_streaming_source_ready), //
avalon_streaming_source.ready
            .valid  (myfifo_shim_0_avalon_streaming_source_valid), //
.valid
            .data   (myfifo_shim_0_avalon_streaming_source_data),  //
.data
            .iready (stin_iready),                                 //
stin.iready
            .ivalid (stin_ivalid),                                 //
.ivalid
            .idata  (stin_idata),                                  //
.idata
```

```
            .clk    (pll_sys_clk_clk)                              //
clk.clk
      );

      lab5_pll pll (
            .ref_clk_clk        (clock_50_clk),            //      ref_clk.clk
            .ref_reset_reset    (reset_reset),             //
ref_reset.reset
            .sys_clk_clk        (pll_sys_clk_clk),         //      sys_clk.clk
            .sdram_clk_clk      (sdram_clk_clk),           //    sdram_clk.clk
            .reset_source_reset (pll_reset_source_reset)   //
reset_source.reset
      );

      lab5_sdram_controller sdram_controller (
            .clk            (pll_sys_clk_clk),
//   clk.clk
            .reset_n        (~rst_controller_001_reset_out_reset),
// reset.reset_n
            .az_addr        (mm_interconnect_0_sdram_controller_s1_address),
//    s1.address
            .az_be_n
(~mm_interconnect_0_sdram_controller_s1_byteenable),   //      .byteenable_n
            .az_cs
(mm_interconnect_0_sdram_controller_s1_chipselect),    //      .chipselect
            .az_data
(mm_interconnect_0_sdram_controller_s1_writedata),     //      .writedata
            .az_rd_n        (~mm_interconnect_0_sdram_controller_s1_read),
//      .read_n
            .az_wr_n        (~mm_interconnect_0_sdram_controller_s1_write),
//      .write_n
            .za_data        (mm_interconnect_0_sdram_controller_s1_readdata),
//      .readdata
            .za_valid
(mm_interconnect_0_sdram_controller_s1_readdatavalid), //      .readdatavalid
            .za_waitrequest
(mm_interconnect_0_sdram_controller_s1_waitrequest),   //      .waitrequest
            .zs_addr        (sdram_addr),
//  wire.export
            .zs_ba          (sdram_ba),
//      .export
            .zs_cas_n       (sdram_cas_n),
//      .export
            .zs_cke         (sdram_cke),
//      .export
            .zs_cs_n        (sdram_cs_n),
//      .export
            .zs_dq          (sdram_dq),
//      .export
            .zs_dqm         (sdram_dqm),
//      .export
            .zs_ras_n       (sdram_ras_n),
//      .export
            .zs_we_n        (sdram_we_n)
//      .export
      );
      spimastershim #(
            .N (16)
      ) spimaster_0 (
            .avs_write
(mm_interconnect_0_spimaster_0_avalon_slave_0_write),     //
avalon_slave_0.write
            .avs_read
(mm_interconnect_0_spimaster_0_avalon_slave_0_read),      //
.read
            .avs_writedata
(mm_interconnect_0_spimaster_0_avalon_slave_0_writedata), //
.writedata
            .avs_readdata
(mm_interconnect_0_spimaster_0_avalon_slave_0_readdata),  //
.readdata
            .clk           (pll_sys_clk_clk),
//          clock.clk
            .reset         (rst_controller_001_reset_out_reset),
//          reset.reset
            .coe_sclk      (spi_sclk),
//  conduit_end_0.sclk
            .coe_mosi      (spi_mosi),
//               .mosi
            .coe_csn       (spi_csn),
//               .csn
            .coe_dcn       (spi_dcn),
//               .dcn
            .coe_resetn    (spi_resetn)
//               .resetn
      );

      lab5_strm_in strm_in (
            .wrclock                          (pll_sys_clk_clk),
//   clk_in.clk
            .reset_n
(~rst_controller_001_reset_out_reset),        // reset_in.reset_n
            .avalonst_sink_valid              (avalon_st_adapter_out_0_valid),
//       in.valid
            .avalonst_sink_data               (avalon_st_adapter_out_0_data),
//         .data
            .avalonst_sink_ready              (avalon_st_adapter_out_0_ready),
//         .ready
            .avalonmm_read_slave_readdata
(mm_interconnect_0_strm_in_out_readdata),    //      out.readdata
            .avalonmm_read_slave_read
(mm_interconnect_0_strm_in_out_read),        //         .read
            .avalonmm_read_slave_address
(mm_interconnect_0_strm_in_out_address),     //         .address
            .avalonmm_read_slave_waitrequest
(mm_interconnect_0_strm_in_out_waitrequest)  //         .waitrequest
      );

      lab5_system_id system_id (
            .clock    (pll_sys_clk_clk),
//           clk.clk
            .reset_n  (~rst_controller_reset_out_reset),
//         reset.reset_n
```

```
            .readdata (mm_interconnect_0_system_id_control_slave_readdata),
// control_slave.readdata
            .address  (mm_interconnect_0_system_id_control_slave_address)
//                .address
      );

      lab5_mm_interconnect_0 mm_interconnect_0 (
            .pll_sys_clk_clk                                (pll_sys_clk_clk),
//                              pll_sys_clk.clk
            .cpu_reset_reset_bridge_in_reset_reset
(rst_controller_reset_out_reset),                              //
cpu_reset_reset_bridge_in_reset.reset
            .spimaster_0_reset_reset_bridge_in_reset_reset
(rst_controller_001_reset_out_reset),                          //
spimaster_0_reset_reset_bridge_in_reset.reset
            .cpu_data_master_address
(cpu_data_master_address),                                     //
cpu_data_master.address
            .cpu_data_master_waitrequest
(cpu_data_master_waitrequest),                                 //
.waitrequest
            .cpu_data_master_byteenable
(cpu_data_master_byteenable),                                  //
.byteenable
            .cpu_data_master_read
(cpu_data_master_read),                                        //
.read
            .cpu_data_master_readdata
(cpu_data_master_readdata),                                    //
.readdata
            .cpu_data_master_write
(cpu_data_master_write),                                       //
.write
            .cpu_data_master_writedata
(cpu_data_master_writedata),                                   //
.writedata
            .cpu_data_master_debugaccess
(cpu_data_master_debugaccess),                                 //
.debugaccess
            .cpu_instruction_master_address
(cpu_instruction_master_address),                              //
cpu_instruction_master.address
            .cpu_instruction_master_waitrequest
(cpu_instruction_master_waitrequest),                          //
.waitrequest
            .cpu_instruction_master_read
(cpu_instruction_master_read),                                 //
.read
            .cpu_instruction_master_readdata
(cpu_instruction_master_readdata),                             //
.readdata
            .cpu_debug_mem_slave_address
(mm_interconnect_0_cpu_debug_mem_slave_address),               //
cpu_debug_mem_slave.address
            .cpu_debug_mem_slave_write
(mm_interconnect_0_cpu_debug_mem_slave_write),                 //
.write
```

```
            .cpu_debug_mem_slave_read
(mm_interconnect_0_cpu_debug_mem_slave_read),                //
.read
            .cpu_debug_mem_slave_readdata
(mm_interconnect_0_cpu_debug_mem_slave_readdata),            //
.readdata
            .cpu_debug_mem_slave_writedata
(mm_interconnect_0_cpu_debug_mem_slave_writedata),           //
.writedata
            .cpu_debug_mem_slave_byteenable
(mm_interconnect_0_cpu_debug_mem_slave_byteenable),          //
.byteenable
            .cpu_debug_mem_slave_waitrequest
(mm_interconnect_0_cpu_debug_mem_slave_waitrequest),         //
.waitrequest
            .cpu_debug_mem_slave_debugaccess
(mm_interconnect_0_cpu_debug_mem_slave_debugaccess),         //
.debugaccess
            .interval_timer_s1_address
(mm_interconnect_0_interval_timer_s1_address),               //
interval_timer_s1.address
            .interval_timer_s1_write
(mm_interconnect_0_interval_timer_s1_write),                 //
.write
            .interval_timer_s1_readdata
(mm_interconnect_0_interval_timer_s1_readdata),              //
.readdata
            .interval_timer_s1_writedata
(mm_interconnect_0_interval_timer_s1_writedata),             //
.writedata
            .interval_timer_s1_chipselect
(mm_interconnect_0_interval_timer_s1_chipselect),            //
.chipselect
            .jtag_uart_avalon_jtag_slave_address
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_address),     //
jtag_uart_avalon_jtag_slave.address
            .jtag_uart_avalon_jtag_slave_write
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_write),       //
.write
            .jtag_uart_avalon_jtag_slave_read
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_read),        //
.read
            .jtag_uart_avalon_jtag_slave_readdata
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_readdata),    //
.readdata
            .jtag_uart_avalon_jtag_slave_writedata
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_writedata),   //
.writedata
            .jtag_uart_avalon_jtag_slave_waitrequest
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_waitrequest), //
.waitrequest
            .jtag_uart_avalon_jtag_slave_chipselect
(mm_interconnect_0_jtag_uart_avalon_jtag_slave_chipselect),  //
.chipselect
            .sdram_controller_s1_address
(mm_interconnect_0_sdram_controller_s1_address),             //
sdram_controller_s1.address
```

```
            .sdram_controller_s1_write
(mm_interconnect_0_sdram_controller_s1_write),              //
.write
            .sdram_controller_s1_read
(mm_interconnect_0_sdram_controller_s1_read),               //
.read
            .sdram_controller_s1_readdata
(mm_interconnect_0_sdram_controller_s1_readdata),           //
.readdata
            .sdram_controller_s1_writedata
(mm_interconnect_0_sdram_controller_s1_writedata),          //
.writedata
            .sdram_controller_s1_byteenable
(mm_interconnect_0_sdram_controller_s1_byteenable),         //
.byteenable
            .sdram_controller_s1_readdatavalid
(mm_interconnect_0_sdram_controller_s1_readdatavalid),      //
.readdatavalid
            .sdram_controller_s1_waitrequest
(mm_interconnect_0_sdram_controller_s1_waitrequest),        //
.waitrequest
            .sdram_controller_s1_chipselect
(mm_interconnect_0_sdram_controller_s1_chipselect),         //
.chipselect
            .spimaster_0_avalon_slave_0_write
(mm_interconnect_0_spimaster_0_avalon_slave_0_write),       //
spimaster_0_avalon_slave_0.write
            .spimaster_0_avalon_slave_0_read
(mm_interconnect_0_spimaster_0_avalon_slave_0_read),        //
.read
            .spimaster_0_avalon_slave_0_readdata
(mm_interconnect_0_spimaster_0_avalon_slave_0_readdata),    //
.readdata
            .spimaster_0_avalon_slave_0_writedata
(mm_interconnect_0_spimaster_0_avalon_slave_0_writedata),   //
.writedata
            .strm_in_out_address
(mm_interconnect_0_strm_in_out_address),                    //
strm_in_out.address
            .strm_in_out_read
(mm_interconnect_0_strm_in_out_read),                       //
.read
            .strm_in_out_readdata
(mm_interconnect_0_strm_in_out_readdata),                   //
.readdata
            .strm_in_out_waitrequest
(mm_interconnect_0_strm_in_out_waitrequest),                //
.waitrequest
            .system_id_control_slave_address
(mm_interconnect_0_system_id_control_slave_address),        //
system_id_control_slave.address
            .system_id_control_slave_readdata
(mm_interconnect_0_system_id_control_slave_readdata)        //
.readdata
      );

      lab5_irq_mapper irq_mapper (
            .clk           (pll_sys_clk_clk),                //       clk.clk
            .reset         (rst_controller_reset_out_reset), //
clk_reset.reset
            .receiver0_irq (irq_mapper_receiver0_irq),       // receiver0.irq
            .receiver1_irq (irq_mapper_receiver1_irq),       // receiver1.irq
            .sender_irq    (cpu_irq_irq)                     //    sender.irq
      );

      lab5_avalon_st_adapter #(
            .inBitsPerSymbol (32),
            .inUsePackets    (0),
            .inDataWidth     (32),
            .inChannelWidth  (0),
            .inErrorWidth    (0),
            .inUseEmptyPort  (0),
            .inUseValid      (1),
            .inUseReady      (1),
            .inReadyLatency  (0),
            .outDataWidth    (32),
            .outChannelWidth (0),
            .outErrorWidth   (0),
            .outUseEmptyPort (0),
            .outUseValid     (1),
            .outUseReady     (1),
            .outReadyLatency (1)
      ) avalon_st_adapter (
            .in_clk_0_clk   (pll_sys_clk_clk),                              //
in_clk_0.clk
            .in_rst_0_reset (rst_controller_001_reset_out_reset),           //
in_rst_0.reset
            .in_0_data      (myfifo_shim_0_avalon_streaming_source_data),  //
in_0.data
            .in_0_valid     (myfifo_shim_0_avalon_streaming_source_valid), //
.valid
            .in_0_ready     (myfifo_shim_0_avalon_streaming_source_ready), //
.ready
            .out_0_data     (avalon_st_adapter_out_0_data),                 //
out_0.data
            .out_0_valid    (avalon_st_adapter_out_0_valid),                //
.valid
            .out_0_ready    (avalon_st_adapter_out_0_ready)                 //
.ready
      );

      altera_reset_controller #(
            .NUM_RESET_INPUTS          (2),
            .OUTPUT_RESET_SYNC_EDGES   ("deassert"),
            .SYNC_DEPTH                (2),
            .RESET_REQUEST_PRESENT     (0),
            .RESET_REQ_WAIT_TIME       (1),
            .MIN_RST_ASSERTION_TIME    (3),
            .RESET_REQ_EARLY_DSRT_TIME (1),
            .USE_RESET_REQUEST_IN0     (0),
            .USE_RESET_REQUEST_IN1     (0),
            .USE_RESET_REQUEST_IN2     (0),
            .USE_RESET_REQUEST_IN3     (0),
            .USE_RESET_REQUEST_IN4     (0),
            .USE_RESET_REQUEST_IN5      (0),
            .USE_RESET_REQUEST_IN6      (0),
            .USE_RESET_REQUEST_IN7      (0),
            .USE_RESET_REQUEST_IN8      (0),
            .USE_RESET_REQUEST_IN9      (0),
            .USE_RESET_REQUEST_IN10     (0),
            .USE_RESET_REQUEST_IN11     (0),
            .USE_RESET_REQUEST_IN12     (0),
            .USE_RESET_REQUEST_IN13     (0),
            .USE_RESET_REQUEST_IN14     (0),
            .USE_RESET_REQUEST_IN15     (0),
            .ADAPT_RESET_REQUEST        (0)
      ) rst_controller (
            .reset_in0      (cpu_debug_reset_request_reset),  //
reset_in0.reset
            .reset_in1      (pll_reset_source_reset),         //
reset_in1.reset
            .clk            (pll_sys_clk_clk),                //
clk.clk
            .reset_out      (rst_controller_reset_out_reset), //
reset_out.reset
            .reset_req      (),                               // (terminated)
            .reset_req_in0  (1'b0),                           // (terminated)
            .reset_req_in1  (1'b0),                           // (terminated)
            .reset_in2      (1'b0),                           // (terminated)
            .reset_req_in2  (1'b0),                           // (terminated)
            .reset_in3      (1'b0),                           // (terminated)
            .reset_req_in3  (1'b0),                           // (terminated)
            .reset_in4      (1'b0),                           // (terminated)
            .reset_req_in4  (1'b0),                           // (terminated)
            .reset_in5      (1'b0),                           // (terminated)
            .reset_req_in5  (1'b0),                           // (terminated)
            .reset_in6      (1'b0),                           // (terminated)
            .reset_req_in6  (1'b0),                           // (terminated)
            .reset_in7      (1'b0),                           // (terminated)
            .reset_req_in7  (1'b0),                           // (terminated)
            .reset_in8      (1'b0),                           // (terminated)
            .reset_req_in8  (1'b0),                           // (terminated)
            .reset_in9      (1'b0),                           // (terminated)
            .reset_req_in9  (1'b0),                           // (terminated)
            .reset_in10     (1'b0),                           // (terminated)
            .reset_req_in10 (1'b0),                           // (terminated)
            .reset_in11     (1'b0),                           // (terminated)
            .reset_req_in11 (1'b0),                           // (terminated)
            .reset_in12     (1'b0),                           // (terminated)
            .reset_req_in12 (1'b0),                           // (terminated)
            .reset_in13     (1'b0),                           // (terminated)
            .reset_req_in13 (1'b0),                           // (terminated)
            .reset_in14     (1'b0),                           // (terminated)
            .reset_req_in14 (1'b0),                           // (terminated)
            .reset_in15     (1'b0),                           // (terminated)
            .reset_req_in15 (1'b0)                            // (terminated)
      );

      altera_reset_controller #(
            .NUM_RESET_INPUTS           (1),
            .OUTPUT_RESET_SYNC_EDGES    ("deassert"),
            .SYNC_DEPTH                (2),
            .RESET_REQUEST_PRESENT     (0),
            .RESET_REQ_WAIT_TIME       (1),
            .MIN_RST_ASSERTION_TIME    (3),
            .RESET_REQ_EARLY_DSRT_TIME (1),
            .USE_RESET_REQUEST_IN0     (0),
            .USE_RESET_REQUEST_IN1     (0),
            .USE_RESET_REQUEST_IN2     (0),
            .USE_RESET_REQUEST_IN3     (0),
            .USE_RESET_REQUEST_IN4     (0),
            .USE_RESET_REQUEST_IN5     (0),
            .USE_RESET_REQUEST_IN6     (0),
            .USE_RESET_REQUEST_IN7     (0),
            .USE_RESET_REQUEST_IN8     (0),
            .USE_RESET_REQUEST_IN9     (0),
            .USE_RESET_REQUEST_IN10    (0),
            .USE_RESET_REQUEST_IN11    (0),
            .USE_RESET_REQUEST_IN12    (0),
            .USE_RESET_REQUEST_IN13    (0),
            .USE_RESET_REQUEST_IN14    (0),
            .USE_RESET_REQUEST_IN15    (0),
            .ADAPT_RESET_REQUEST       (0)
      ) rst_controller_001 (
            .reset_in0      (pll_reset_source_reset),             //
reset_in0.reset
            .clk            (pll_sys_clk_clk),                    //
clk.clk
            .reset_out      (rst_controller_001_reset_out_reset), //
reset_out.reset
            .reset_req      (),                                   //
(terminated)
            .reset_req_in0  (1'b0),                               //
(terminated)
            .reset_in1      (1'b0),                               //
(terminated)
            .reset_req_in1  (1'b0),                               //
(terminated)
            .reset_in2      (1'b0),                               //
(terminated)
            .reset_req_in2  (1'b0),                               //
(terminated)
            .reset_in3      (1'b0),                               //
(terminated)
            .reset_req_in3  (1'b0),                               //
(terminated)
            .reset_in4      (1'b0),                               //
(terminated)
            .reset_req_in4  (1'b0),                               //
(terminated)
            .reset_in5      (1'b0),                               //
(terminated)
            .reset_req_in5  (1'b0),                               //
(terminated)
            .reset_in6      (1'b0),                               //
(terminated)
            .reset_req_in6  (1'b0),                               //
(terminated)
```

```
            .reset_in7      (1'b0),                              //
(terminated)
            .reset_req_in7  (1'b0),                              //
(terminated)
            .reset_in8      (1'b0),                              //
(terminated)
            .reset_req_in8  (1'b0),                              //
(terminated)
            .reset_in9      (1'b0),                              //
(terminated)
            .reset_req_in9  (1'b0),                              //
(terminated)
            .reset_in10     (1'b0),                              //
(terminated)
            .reset_req_in10 (1'b0),                              //
(terminated)
            .reset_in11     (1'b0),                              //
(terminated)
            .reset_req_in11 (1'b0),                              //
(terminated)
            .reset_in12     (1'b0),                              //
(terminated)
            .reset_req_in12 (1'b0),                              //
(terminated)
            .reset_in13     (1'b0),                              //
(terminated)
            .reset_req_in13 (1'b0),                              //
(terminated)
            .reset_in14     (1'b0),                              //
(terminated)
            .reset_req_in14 (1'b0),                              //
(terminated)
            .reset_in15     (1'b0),                              //
(terminated)
            .reset_req_in15 (1'b0)                               //
(terminated)
      );

endmodule
```

10. C program file

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <unistd.h>

#define MAX_LEVEL 4096                                // Max level from ADC
#define ROWS 64                                       // Rows of display
#define COLUMNS 96                                    // Columns of display
#define BAR_WIDTH 5                                   // Width of the bar
#define FLAG_VALUE 65536                              // 2^16
#define SIN_2PI_16 0.38268343236508978
#define SIN_4PI_16 0.707106781186547460
#define SIN_6PI_16 0.923879532511286740
```

```c
#define C_P_S_2PI_16 1.30656296487637660
#define C_M_S_2PI_16 0.54119610014619690
#define C_P_S_6PI_16 1.3065629648763766
#define C_M_S_6PI_16 -0.54119610014619690

void clean ();
void fill_row(int column, int value);
void R16SRFFT(float input[16], int output[16]);

// colors
unsigned ON[16] = { 0x300f, 0x001f, 0x013ff, 0x013ff, 0x01ed, 0x07e0, 0x01ed,
0x01ed,0x7f01, 0x01ed, 0xff81, 0x01ed, 0xfd60, 0xff60, 0xfa20, 0xf800};

int display [ROWS] [COLUMNS];                              // 2-D array

int initdata[38] = { 0xAE, 0x81, 0xFF, 0x82, 0xFF, 0x83, 0xFF,
    0x87, 0x06, 0x8A, 0x64, 0x8B, 0x78, 0x8C,
    0x64, 0xA0, 0x73, 0xA1, 0x00, 0xA2, 0x00,
    0xA4, 0xA8, 0x3F, 0xAD, 0x8E, 0xB0, 0x00,
    0xB1, 0x31, 0xB3, 0xF0, 0xBB, 0x3A, 0xBE,
    0x3E, 0x2E, 0xAF } ;

#include "system.h"            /* peripheral base addresses */
#define SPIWRITE(x)  (*(int*)SPIMASTER_0_BASE) = (x)
#define SPIREAD (*(int*)SPIMASTER_0_BASE)
#define LEDSET(x)  (*(int*)PIO_BASE) = (x)

int main()
{
    // controller initialization

    int i, j ;

    for ( i=0 ; i<38 ; i++ ) {
        SPIWRITE(initdata[i] | 0x100) ;
        while ( ( SPIREAD & 1 ) == 0 );
    }

    int n, value;
    float input[16] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
      long score[16];
      int output[16];
      int r = 0;
      int data;
      int *padc  = (int*) STRM_IN_BASE ;

      printf("ELEX 7660 Lab 5\n");
      value = 0 ;
      n = 0;

      while(1)
      {
            while(data>>16 != 0)
            {
                  data = padc[0];
            }
            for(n = 0; r != 1 ; n++)
            {
                  n = data>>16;
                  value = data - (n << 16);
                  score[n] = (value)>>4;
                  input[n] = score[n];
                  printf ("x=%d ", score[n]) ;
                  r = (n == 15)? 1:0;
                  data = padc[0];
            }
            R16SRFFT(input, output);                  // FFT
            clean();                                        // Clears array
            for(n = 0; n <16; n++)
            {
                  fill_row(n+1, output[n]);           // fills the array
            }


      // fill framebuffer

            for ( j=0 ; j<COLUMNS ; j++ )
            {
                  for ( i=0 ; i< ROWS ; i++ )
                  {
                        SPIWRITE( display[i][j] >> 8 & 0xFF )   ; /* MS byte
*/
                        while ( ( SPIREAD & 1 ) == 0 );
                        SPIWRITE( display[i][j] & 0xFF ) ;  /* LS byte */
                        while ( ( SPIREAD & 1 ) == 0 );
                  }
            }
            r = 0;
            n = 0;
            usleep(500);
      }
    return 0;
}

void clean()
{
    int i,j;
    for(i = 0; i < ROWS; i++)
    {
        for(j = 0; j < COLUMNS ; j++)
        {
            display [i][j] = 0x0000;
        }
        j = 0;
    }
}

void fill_row(int column, int value)
{
    int i,j;
    if (value > (MAX_LEVEL -4))
    {
      value = (MAX_LEVEL -4);                         // limit of level
    }
    for(i = 0; i < BAR_WIDTH; i++)
    {

        for(j = 0; j < (value * ROWS)/MAX_LEVEL ; j++)
        {
            display[ROWS - (j + 1)][(column -1) * (BAR_WIDTH + 1) + i] =
ON[column - 1];   // turn on the pixel
        }
        j = 0;
    }
}
    //----------------- FFT --------------------------------

    /* INPUT: float input[16], float output[16] */
    /* OUTPUT: none */
    /* EFFECTS:  Places the 16 point fft of input in output in a strange */
    /* order using 10 real multiplies and 79 real adds. */
    /* Re{F[0]}= out0 */
    /* Im{F[0]}= 0 */
    /* Re{F[1]}= out8 */
    /* Im{F[1]}= out12 */
    /* Re{F[2]}= out4 */
    /* Im{F[2]}= -out6 */
    /* Re{F[3]}= out11 */
    /* Im{F[3]}= -out15 */
    /* Re{F[4]}= out2 */
    /* Im{F[4]}= -out3 */
    /* Re{F[5]}= out10 */
    /* Im{F[5]}= out14 */
    /* Re{F[6]}= out5 */
    /* Im{F[6]}= -out7 */
    /* Re{F[7]}= out9 */
    /* Im{F[7]}= -out13 */
    /* Re{F[8]}= out1 */
    /* Im{F[8]}=0 */
    /* F[9] through F[15] can be found by using the formula */
    /* Re{F[n]}=Re{F[(16-n)mod16]} and Im{F[n]}= -Im{F[(16-n)mod16]} */

    /* Note using temporary variables to store intermediate computations */
    /* in the butterflies might speed things up.  When the current version */
    /* needs to compute a=a+b, and b=a-b, I do a=a+b followed by b=a-b-b.  */
    /* So practically everything is done in place, but the number of adds */
    /* can be reduced by doinc c=a+b followed by b=a-b. */

    /* The algorithm behind this program is to find F[2k] and F[4k+1] */
    /* seperately.  To find F[2k] we take the 8 point Real FFT of x[n]+x[n+8]
*/
    /* for n from 0 to 7.  To find F[4k+1] we take the 4 point Complex FFT of
*/
    /* exp(-2*pi*j*n/16)*{x[n] - x[n+8] + j(x[n+12]-x[n+4])} for n from 0 to
3.*/

    void R16SRFFT(float input[16], int output[16]) {
        float temp, out0, out1, out2, out3, out4, out5, out6, out7, out8;
        float out9, out10, out11, out12, out13, out14, out15;
        float out_temp0, out_temp1, out_temp2, out_temp3, out_temp4,
out_temp5;
        float out_temp6, out_temp7, out_temp8, out_temp9, out_temp10,
out_temp11;
        float out_temp12, out_temp13, out_temp14, out_temp15;

        out0 = input[0] + input[8]; /* output[0 through 7] is the data that
we */
        out1 = input[1] + input[9]; /* take the 8 point real FFT of. */
        out2 = input[2] + input[10];
        out3 = input[3] + input[11];
        out4 = input[4] + input[12];
        out5 = input[5] + input[13];
        out6 = input[6] + input[14];
        out7 = input[7] + input[15];

        out8 = input[0] - input[8];   /* inputs 8,9,10,11 are */
        out9 = input[1] - input[9];   /* the Real part of the */
        out10 = input[2] - input[10]; /* 4 point Complex FFT inputs.*/
        out11 = input[3] - input[11];
        out12 = input[12] - input[4]; /* outputs 12,13,14,15 are */
        out13 = input[13] - input[5]; /* the Imaginary pars of  */
        out14 = input[14] - input[6]; /* the 4 point Complex FFT inputs.*/
        out15 = input[15] - input[7];

        /*First we do the "twiddle factor" multiplies for the 4 point CFFT */
        /*Note that we use the following handy trick for doing a complex */
        /*multiply:  (e+jf)=(a+jb)*(c+jd) */
        /*   e=(a-b)*d + a*(c-d)   and    f=(a-b)*d + b*(c+d)  */

        /* C_M_S_2PI/16=cos(2pi/16)-sin(2pi/16) when replaced by
macroexpansion */
        /* C_P_S_2PI/16=cos(2pi/16)+sin(2pi/16) when replaced by
macroexpansion */
        /* (SIN_2PI_16)=sin(2pi/16) when replaced by macroexpansion */
        temp = (out13 - out9)*(SIN_2PI_16);
        out9 = out9*(C_P_S_2PI_16)+temp;
        out13 = out13*(C_M_S_2PI_16)+temp;

        out14 *= (SIN_4PI_16);
        out10 *= (SIN_4PI_16);
        out14 = out14 - out10;
        out10 = out14 + out10 + out10;

        temp = (out15 - out11)*(SIN_6PI_16);
        out11 = out11*(C_P_S_6PI_16)+temp;
        out15 = out15*(C_M_S_6PI_16)+temp;

        /* The following are the first set of two point butterfiles */
        /* for the 4 point CFFT */

        out8 += out10;
        out10 = out8 - out10 - out10;

        out12 += out14;
        out14 = out12 - out14 - out14;
        out9 += out11;
        out11 = out9 - out11 - out11;

        out13 += out15;
        out15 = out13 - out15 - out15;

        /*The followin are the final set of two point butterflies */
        out_temp1 = out8 + out9;
        out_temp7 = out8 - out9;

        out_temp9 = out12 + out13;
        out_temp15 = out13 - out12;

        out_temp5 = out10 + out15;        /* implicit multiplies by */
        out_temp13 = out14 - out11;        /* a twiddle factor of -j */
        out_temp3 = out10 - out15;  /* implicit multiplies by */
        out_temp11 = -out14 - out11;  /* a twiddle factor of -j */


        /* What follows is the 8-point FFT of points output[0-7] */
        /* This 8-point FFT is basically a Decimation in Frequency FFT */
        /* where we take advantage of the fact that the initial data is
real*/
        /* First set of 2-point butterflies */

        out0 = out0 + out4;
        out4 = out0 - out4 - out4;
        out1 = out1 + out5;
        out5 = out1 - out5 - out5;
        out2 += out6;
        out6 = out2 - out6 - out6;
        out3 += out7;
        out7 = out3 - out7 - out7;

        /* Computations to find X[0], X[4], X[6] */

        out_temp0 = out0 + out2;
        out_temp4 = out0 - out2;
        out1 += out3;
        out_temp12 = out3 + out3 - out1;

        out_temp0 += out1;  /* Real Part of X[0] */
        out_temp8 = out_temp0 - out1 - out1; /*Real Part of X[4] */
        /* out2 = Real Part of X[6] */
        /* out3 = Imag Part of X[6] */
        /* Computations to find X[5], X[7] */

        out5 *= SIN_4PI_16;
        out7 *= SIN_4PI_16;
        out5 = out5 - out7;
        out7 = out5 + out7 + out7;

        out_temp14 = out6 - out7; /* Imag Part of X[5] */
        out_temp2 = out5 + out4; /* Real Part of X[7] */
        out_temp6 = out4 - out5; /*Real Part of X[5] */
        out_temp10 = -out7 - out6; /* Imag Part of X[7] */
        // Magnitude of the output
        // Modified section
        // Gets magnitude
        output[0] = abs(out_temp0);
        output[1] = 2*sqrtf(abs(out_temp1*out_temp1) + abs(out_temp9 *
out_temp9));
        output[2] = 6*sqrtf(abs(out_temp2*out_temp2) + abs(out_temp10 *
out_temp10));
        output[3] = 4*sqrtf(abs(out_temp3*out_temp3) + abs(out_temp11 *
out_temp11));
        output[4] = 5*sqrtf(abs(out_temp4*out_temp4) + abs(out_temp12 *
out_temp12));
        output[5] = 5*sqrtf(abs(out_temp5*out_temp5) + abs(out_temp13 *
out_temp13));
        output[6] = 6*sqrtf(abs(out_temp6*out_temp6) + abs(out_temp14 *
out_temp14));
        output[7] = 2*sqrtf(abs(out_temp7*out_temp7) + abs(out_temp15 *
out_temp15));
        output[8] = 5*abs(out_temp8);
        output[9] = 5*output[7];
        output[10] = 2*output[6];
        output[11] = 2*output[5];
        output[12] = 2*output[4];
        output[13] = 2*output[3];
        output[14] = sqrtf(abs(out_temp2*out_temp2) + abs(out_temp9 *
out_temp9));
        output[15] = sqrtf(abs(out_temp1*out_temp1) + abs(out_temp8 *
out_temp8));

    }
```