|  |
| --- |
| Design and Test of Microelectronic Systems SS\_19 |
|  |
| Lab 1: Digital Audio Filter on Programmable Logic |
| Report submitted by :  **Bharath Ramachandraiah**  Matrikelnr : 762596  **Bhalaji Virinchipuram Magendran**  Matrikelnr : 762628 |
|  |
| **May/10/2019** |

Content

[1 Introduction 3](#_Toc514104348)

[2 Objective 3](#_Toc514104349)

[3 Assignment 3](#_Toc514104350)

[3.1 Task 1: I2S Data Format for Audio Exchange 3](#_Toc514104351)

[3.2 Task 2: Block diagram of hierarchical design 4](#_Toc514104352)

[3.3 Task 3: Extending the functionality of the design: 5](#_Toc514104353)

[3.4 Task 4: Synthesis and Utilization report: 7](#_Toc514104354)

[3.5 Task 5: Verifying the Design: 9](#_Toc514104355)

# Introduction

The hardware used was the ZedBoard with an ARM SoC being the Processing System and a Programmable Logic block.

On the ZedBoard, the ADI chip ADAU1761 codec is dedicated for audio, it is equipped with two 24bit ADCs and 24bit DACs with a wide range of supported sampling rate from 8 kHz to 96 kHz. The audio interface for ZedBoard connects the ADAU1761 audio codec on the ZedBoard to Zynq PL. Audio signals can be received in stereo from the line in jack and transmitted to the headphone out jack. Three different filters are designed and these are selected using the slides switches on the ZedBoard.

# Objective

The objective is to route the off-board peripherals to the programmable logic PL. In this exercise, a seven segment display is connected to Pmods of the ZedBoard. Parallel processing in FPGAs is demonstrated by testing the implementation of algorithms for Digital filters.

# Assignment

## Task 1: I2S Data Format for Audio Exchange

**a).**

Task-Description:

Explain the data format I2S for audio data exchange between Audio Codec and Zynq device! Name data signals and clock signals of this data format by studying ADAU1761 product description!

Elaboration:

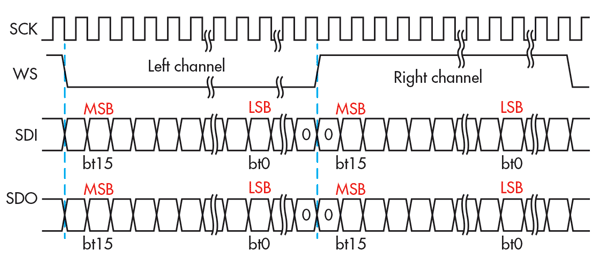


Figure 1: Data Format of I2S

The I²S protocol outlines one specific type of PCM digital audio communication with defined parameters outlined in the Philips specification.

The bus consists of at least three lines:

1. Bit clock line:  
   Continuous serial clock (SCK). Typically written "bit clock (BCLK)"
2. Word clock line:  
   Word select (WS). Typically called "left-right clock (LRCLK)". 0 = Left channel, 1 = Right channel, Frame Sync (FS)
3. At least one multiplexed data line:   
   serial data (SD), but can be called SDATA, SDIN, SDOUT, DACDAT, ADCDAT, etc.

## Task 2: Block diagram of hierarchical design

**(a).**

Task-Description:

Generate the RTL schematic of the highest instance *adau1761\_test.vhd*.

Elaboration:

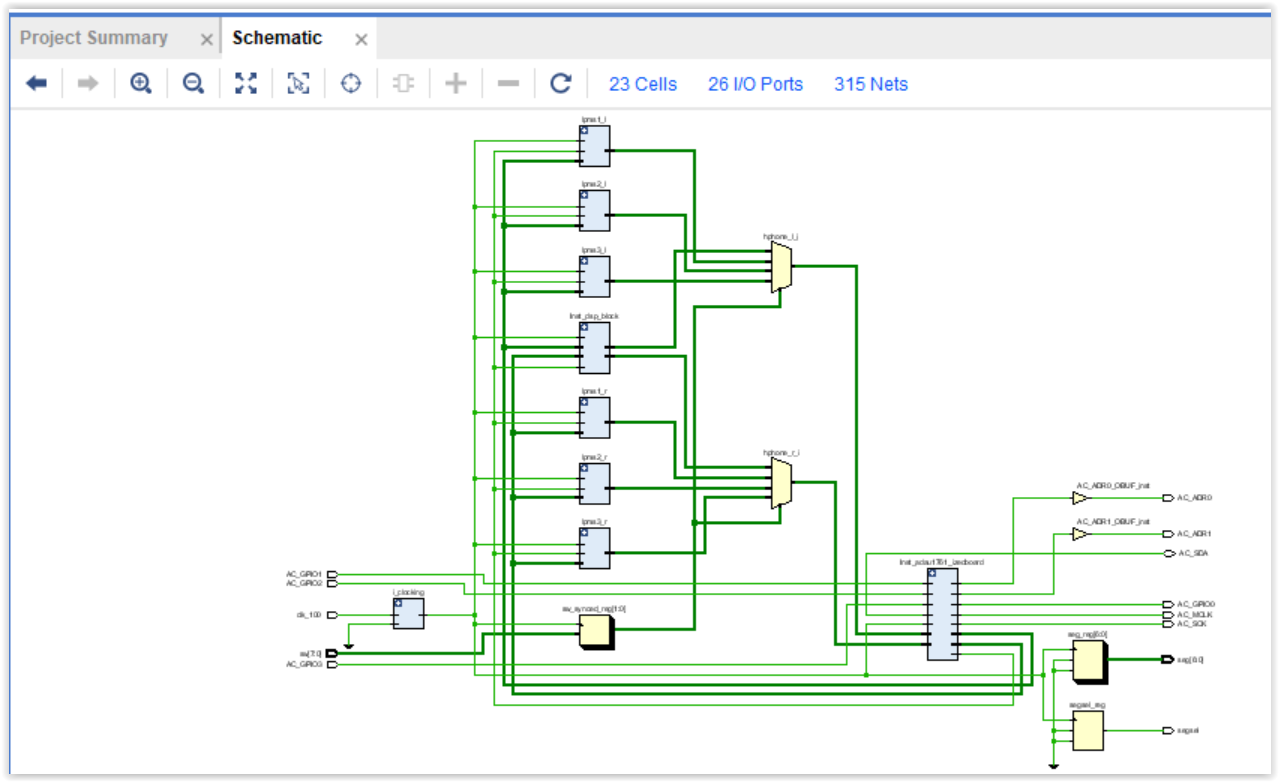


Figure 2: RTL Schematic of adau1761.vhd

**(b) Description of the Output signals**

Task-Description:

Answer the following questions based on the RTL Schematic of adau1761.vhd.

Elaboration:

1. What is the purpose of the multiplexer hphone\_l\_i and hphone\_r\_i?

* These multiplexers are used to select type of filtering among the following four, i.e, Null- filter, 8-point average, 16-point average, 32-point average.

1. What is the function of Inst\_dsp\_block?

* Inst\_dsp\_block is the NULL-filter which is used to send the audio input to output without any filtering.

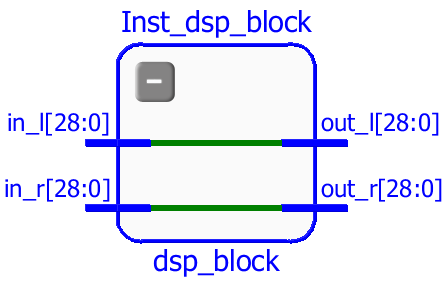


Figure 3: Inst\_dsp\_block

1. Which port is connected to the serial data output of ADC?

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | Description | Zynq Pin | ADAU1761 pin |
| AC-GPIO1 | Digital Audio Serial Data ADC Output | AA7 | 26 |

1. Which port is connected to the serial data input of DAC?

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | Description | Zynq Pin | ADAU1761 pin |
| AC-GPIO0 | Digital Audio Serial Data DAC Input | Y8 | 27 |

1. What is the clock frequency of MCLK? What is the sample frequency of the ADC and DAC?

* MCLK = 24 MHz

ADC\_CLK = 48 KHz

DAC\_CLK = 48 KHz

1. What is the filter length N of the three low pass filters? Calculate the cutoff frequency of each filter!

* 8, 16 and 32 are the filter lengths of three low pass filters.
* Cutoff frequencies,
  + N= 8, F\_cutoff = 6 KHz
  + N = 16, F\_cutoff = 3 KHz
  + N = 32, F\_cutoff = 1.5 KHz

## Task 3: Extending the functionality of the design:

Task-Description:

1. To display the selected filter number on the attached seven segment display. The seven segment display will be attached to the Pmod of the ZedBoard.
2. To add the I/O port settings for communicating to the Pmods.

Elaboration:

The configuration of the seven segment is as shown below .The bit pattern for displaying the number corresponding to each filter is identified and the corresponding changes are made in the code.

****

Figure 4:Seven Segment Display Connections

I/O Port Configuration:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Direction** | **Site** | **I/O Standard** |
| Seg[0] | Output | Y11 | LVCMOS33 |
| Seg[1] | Output | AA11 | LVCMOS33 |
| Seg[2] | Output | Y10 | LVCMOS33 |
| Seg[3] | Output | AA9 | LVCMOS33 |
| Seg[4] | Output | W12 | LVCMOS33 |
| Seg[5] | Output | W11 | LVCMOS33 |
| Seg[6] | Output | V10 | LVCMOS33 |

The code snippet is as shown below.

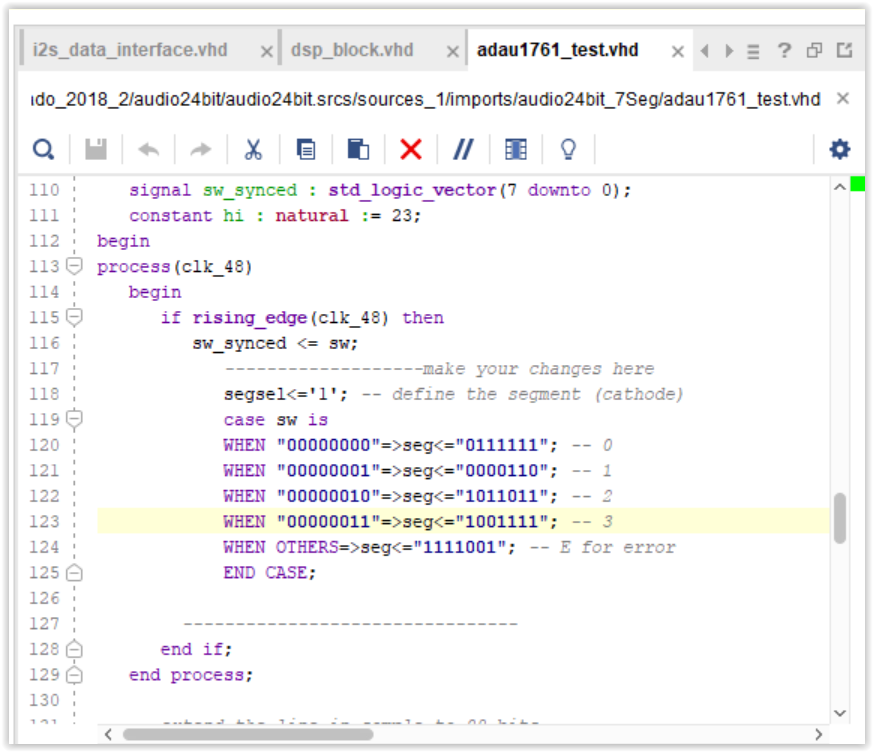


Figure 5:Implementation of the seven segment display

**Code Comments :**

WHEN “0000000”=>seg<=”0111111” ; // Seven Segment o/p to display 0

WHEN “0000001”=> seg <=”0000110” ; // Seven Segment o/p to display 1

WHEN “0000010”=> seg <=”1011011” ; // Seven Segment o/p to display 2

WHEN “0000011”=> seg <=”1001111” ; // Seven Segment o/p to display 3

WHEN OTHERS=> seg<=”1111001” ; // Seven Segment o/p to display E for Error

## Task 4: Synthesis and Utilization report:

Task-Description:

To synthesis the design and to analyze the utilization report and explain about Slice LUTs, Slice Register, Block RAM and DSPs.

Elaboration:

* 1. ***Slice LUT***: A look up table (LUT) is a collection of logic gates hard-wired on the FPGA. LUTs store a predefined list of outputs for every combination of inputs and provide a fast way to retrieve information out of a logic operation. A number of LUTs together is called a “Slice LUT”.
  2. ***Slice Register***: A register is a group of flip-flops that stores a bit pattern. A register on the FPGA has a clock, input data, output data and enable signal port. Every clock cycle, the input data is latched, stored internally, and the output data is matched to store the internally stored data. The LUT and Register utilization in this design is as shown below.

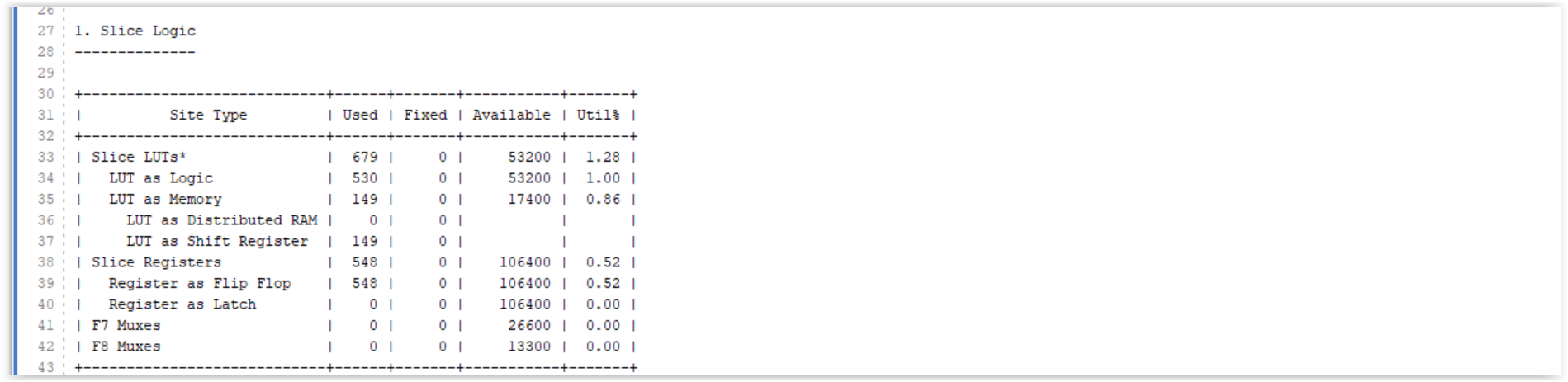


Figure 6:Utilization of Slice Registers and Slice LUT

* 1. ***Block RAM:*** Block RAM is the memory that is embedded throughout the FPGA for storing data. In general, block RAM is used when synthesizing memory and FIFO functions. The RAM utilization in this design is as shown below.

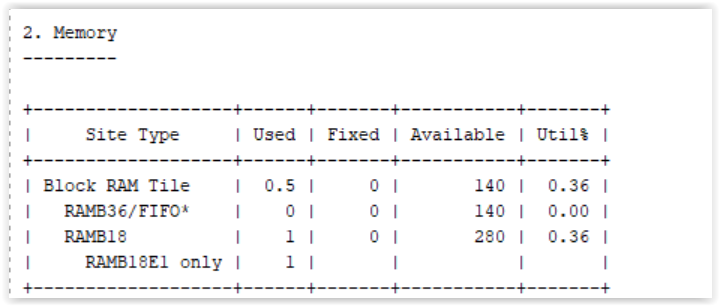


Figure 7:Utilization of Block RAM

* 1. ***DSPs***: These blocks are used in implementing complex arithmetic calculations at a faster rate than the normal logic blocks. The DSP blocks utilization in this design is as shown below.

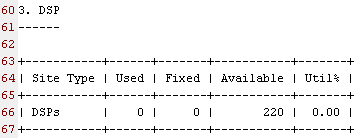


Figure 8:Utilization of DSP Blocks

## Task 5: Verifying the Design:

Task-Description:

To verify if the right filter number is displayed on the seven segment display.

Elaboration:

The filters are selected using a Mux which is controlled through the switches on the ZedBoard. SW [1] and SW [0] are used to select among three filters. The pattern shown in the table below is the expected filter number for each combination of the switch SW.

|  |  |  |
| --- | --- | --- |
| SW[1] | SW[0] | Filter Number |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |

**Result:**

The output on the seven segment display matched the combination of the input switches and the design was verified.

