Digital System Design

SS 2022

Lab 2: **Parallel Processing (Digital Audio Filter)**

Report submitted by:

Charan Krishnamurthy
Matrikelnr:1112311

Kolvi Nayan Mallikarjuna
Matrikelnr: 1112310

01-07-2022

Contents

1 Introduction

The parallelism and customizable architecture inherent in the FPGA architecture is ideal for high-throughput processing. Digital filters are one possible application. In this lab we are dealing with a digital audio filter implemented on the **Programmable Logic (PL)** of the SoC ZYNQ device.

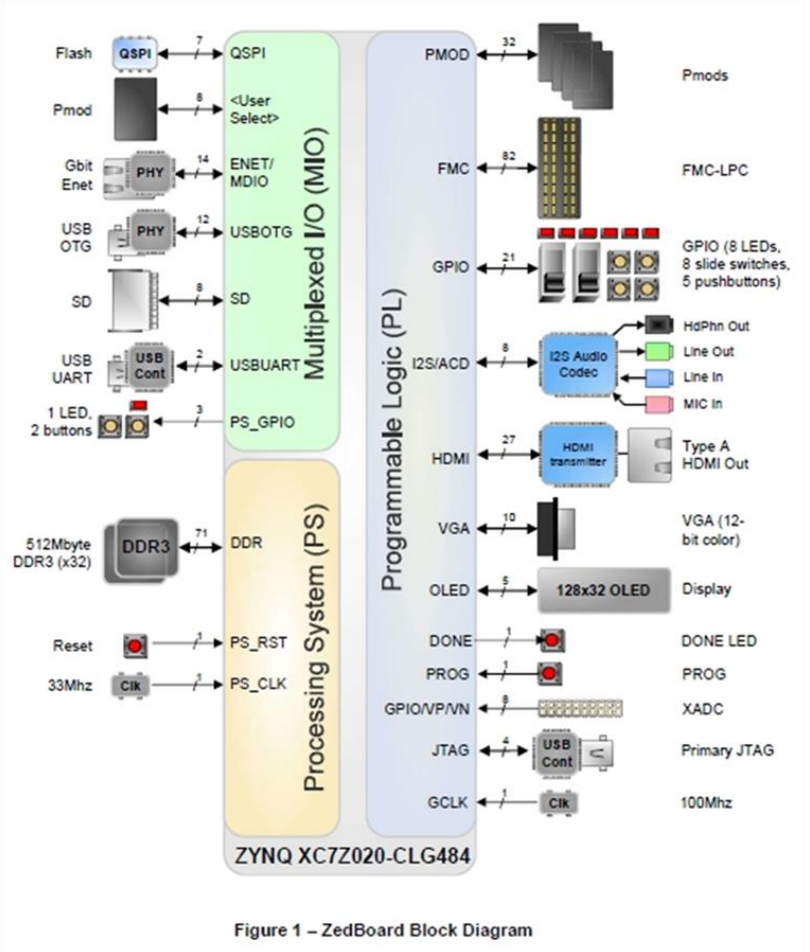


Figure 1 – ZedBoard Block Diagram

Figure 1: ZenBoard Block Diagram

Analog input is the Line In and analog output is the Headphone. The on-board Audio Codec is used for AD und DA conversion. In this project the user can select between 3 different filters using the slides switches on ZedBoard.

2 Implementation

Topic a – I2S data format for audio exchange

Task-Description:

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

Analysis:

I²S is an electrical serial interface standard used for connecting digital audio devices together. It is used to communicate pulse code modulation audio data between integrated circuits in an electronic device. The I²S bus separates clock and serial data signals, resulting in simpler receivers than those required for asynchronous communications systems that need to recover the clock from the data stream.

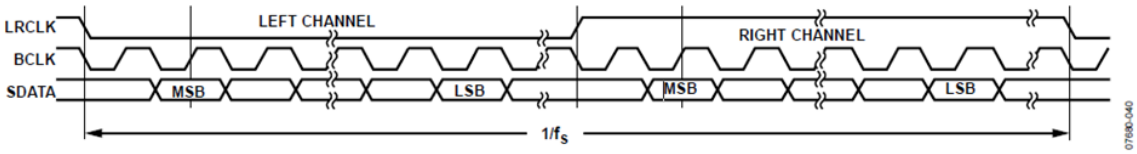


Figure 58. I²S Mode—16 Bits to 24 Bits per Channel

Figure 2: I2S Mode

The bus consists of at least three lines:

1. Bit clock line
    a. Officially "continuous serial clock (SCK)". Typically written "bit clock (BCLK)".
2. Word clock line
    a. Officially "word select (WS)". Typically called "left-right clock (LRCLK)" or "frame sync (FS)".
    b. 0 = Left channel, 1 = Right channel
3. At least one multiplexed data line
    a. Officially "serial data (SD)", but can be called SDATA, SDIN, SDOUT, DACDAT, ADCDAT, etc.

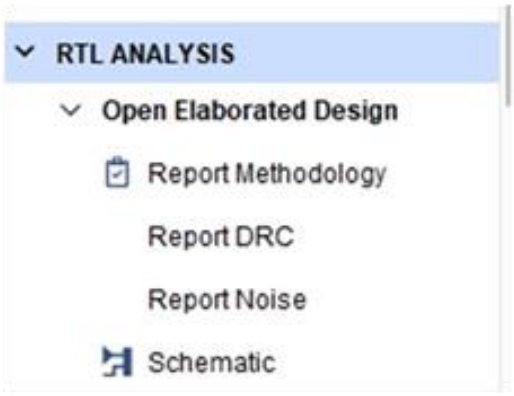
Pin	Pnemonic	Type	Description
26	ADC_SDATA/GPIO1	D_IO	ADC Serial Output Data (ADC_SDATA)
27	DAC_SDATA/GPIO0	D_IO	DAC Serial Input Data (DAC_SDATA).
28	BCLK/GPIO2	D_IO	Serial Data Port Bit Clock (BCLK).
29	LRCLK/GPIO3	D_IO	Serial Data Port Frame Clock (LRCLK).

Table 1: Data IO Description

Topic b – Information about audio24bit Project

Task-Description:

Open project file audio24bit in Vivado. Generate the RTL Schematic of the highest instance adau1761_test.vhd: In Flow Navigator click on RTL ANALYSIS/Open Elaborated Design, then click on Schematic.



Make a screenshot of the schematic!

Analysis:

Schematic:

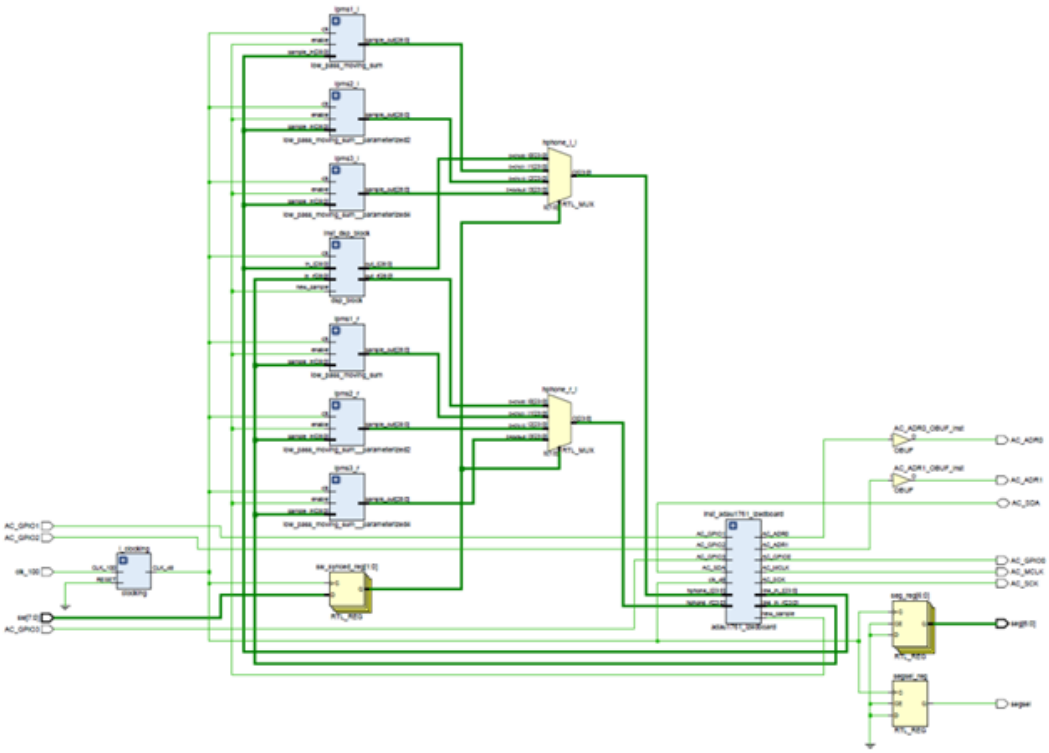


Figure 3: Schematic diagram of adau1761

- What is the purpose of the multiplexer hphone_l_i and hphone_r_i?
Answer: hphone_l_i and hphone_r_i are used to select between signals out of 4 different filters i.e., Null Filter, 8 Point Average, 16 Point Average and 32 Point Average.

- What is the function of Inst_dsp_block?
Answer: Inst_dsp_block acts as a Null Filter for the audio signal.

- Which port is connected to the serial data output of ADC?
Answer: ADC_SDATA/GPIO1 (pin 26) is connected to data output of ADC.

- Which port is connected to the serial data input of DAC?
Answer: DAC_SDATA/GPIO0 (pin 27) is connected to data output of DAC.

- Which clock frequency is used for signal MCLK?
Answer: External master clock uses usually 24MHz or 12MHz.

- What is the filter length N of the three low pass filters?
Answer: N is 8, 16 and 32 respectively.

- The low pass filters are running-sum FIR filter. Calculate the cut-off frequency of each filter!

Answer: For N Values of 8, 16 and 32 Cut-off Frequencies are 6 KHz, 3 KHz and 1.5 KHz respectively. (Cut-off frequencies = Fs/N).

- Calculate the pre-configured sample frequency of the ADC and DAC (refer to audio codec ADAU1762 product description, chapter: clocking and sampling rates and to project source file for configuration of audio codec)!

Answer:

The ADCs, DACs, and serial port share a common sampling rate that is set in Register R17 Converter Control 0 register, Address 0x4017). The core clock can be derived directly from MCLK or it can be generated by the PLL.

For example, if the input to CLKSRC = 49.152 MHz (from PLL), then

- *INFREQ[1:0]* = 1024 × *fs*
- *fs* = 49.152 MHz/1024 = 48 kHz

where, fs is the sampling frequency.

Topic c – Top Level Input and Output Signals

Task-Description:

Make a print out of the RTL schematic and explain each top-level input and output!

Analysis:

Configuration

Name	Direction	Package Pin
Ports (26)		
sw (8)	IN	
segsel	OUT	
seg (7)	OUT	
clk_100	IN	Y9
AC_SDA	INOUT	AB5
AC_SCK	OUT	AB4
AC_MCLK	OUT	AB2
AC_GPIO3	IN	Y6
AC_GPIO2	IN	AA6
AC_GPIO1	IN	AA7
AC_GPIO0	OUT	Y8
AC_ADR1	OUT	Y5
AC_ADR0	OUT	AB1

Figure 4: RTL schematic adau1761

Description:

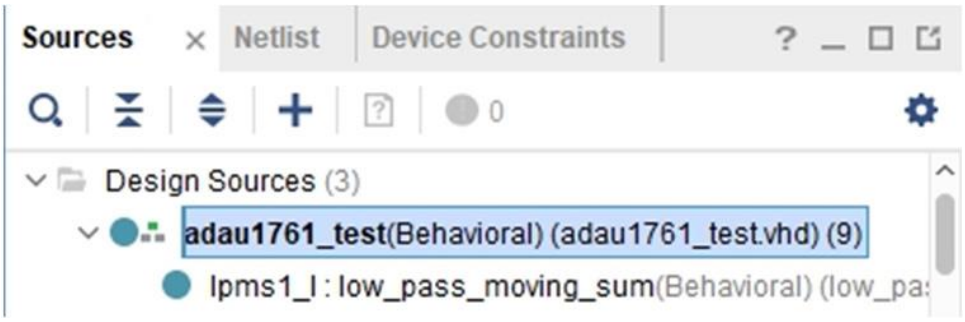
Signal Name	Direction	Description
sw[0:7]	IN	Filter select lines.
AC_ADR0	OUT	I2C Address Bit 0/SPI Latch Signal
AC_ADR1	OUT	I2C Address Bit 1/SPI Data Input
AC_GPIO0	OUT	DAC Serial Input Data (DAC_SDATA).
AC_GPIO1	IN	ADC Serial Output Data (ADC_SDATA)
AC_GPIO2	IN	Serial Data Port Bit Clock (BCLK).
AC_GPIO3	IN	Serial Data Port Frame Clock (LRCLK).
AC_MCLK	OUT	Master clock input.
AC_SCK	OUT	I2C Data (SDA). This pin is a bidirectional open-collector input/output. The line connected to this pin should have a 2 kΩ pull-up resistor.
AC_SDA	INOUT	I2C Serial Data Interface
clk_100	IN	100 MHz Input Clock.
Seg [0:6]	OUT	Seven Segment Connection
segsel	OUT	Select Seven Segment Display

Table 2: Port description

Topic d – Functionality extension of Switch

Task-Description:

Extend the functionality of the design: Display selected filter number on the attached 7-segment display. The 7-segment display is getting attached to the Pmod JA1+JB1 of ZedBoard.Open VHDL code adau1761_test.vhd: In Sources tab double click on adau1761_test.



The part of the VHDL code you need to change is already highlighted. You also need to add I/O port settings for the communication to the Pmods. Table of I/O opens, edit column Package Pin, I/O Std. for the missing ports to control7-segment display (refer to ZedBoard user manual).

Code:

```
process(clk_48)
   begin
      if rising_edge(clk_48) then
         sw_synced <= sw;
            ------------------make your changes here
            segsel<='1'; -- define the segment (cathode)
            case sw is
            WHEN "00000000"=>seg<="0111111"; -- control 7-Segment
            WHEN "00000001"=>seg<="0000110";
            WHEN "00000010"=>seg<="1011011";
            WHEN "00000011"=>seg<="1001111";
            WHEN OTHERS=>seg<="1111001";
            END CASE;

          --------------------------------
      end if;
   end process;
```

Figure 5: Modified code for Button selection

7-Segment Display Configuration:

Name	Direction	Package Pin	I/O Std	Vcco
Ports (26)				
seg (7)	OUT		LVCMOS33*	3.300
seg[4]	OUT	W12	LVCMOS33*	3.300
seg[1]	OUT	AA11	LVCMOS33*	3.300
seg[6]	OUT	V10	LVCMOS33*	3.300
seg[2]	OUT	Y10	LVCMOS33*	3.300
seg[5]	OUT	W11	LVCMOS33*	3.300
seg[3]	OUT	AA9	LVCMOS33*	3.300
seg[0]	OUT	Y11	LVCMOS33*	3.300

Table 3: Port Assignment and description for 7 Segment Display

Output:

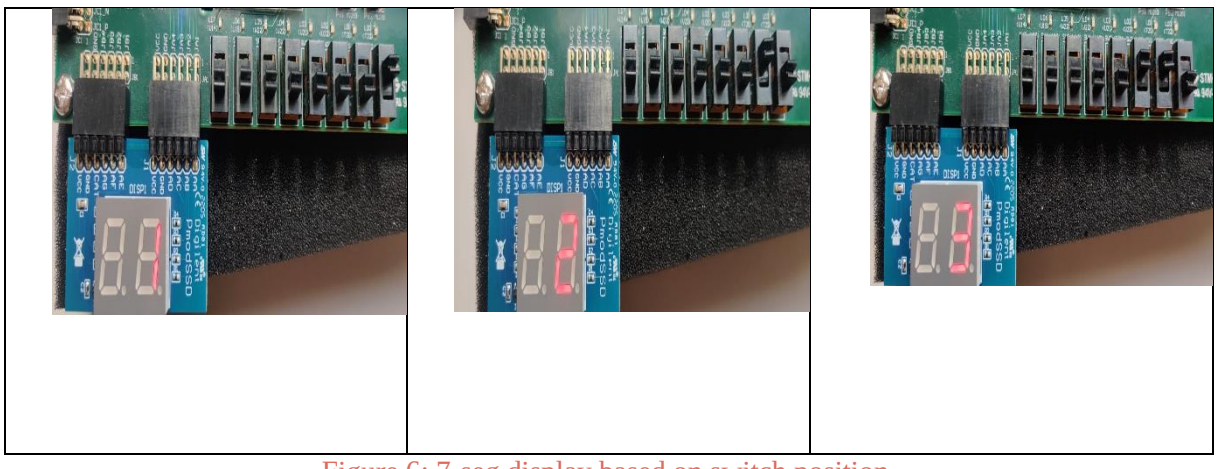


Figure 6: 7-seg display based on switch position.

Topic e – Synthesis and Report Analysis

Task-Description:

Open Report on utilization of resources of Synth Design. Do research on Zynq device. What is the meaning of Slice LUTs, Slice Register, Block RAM, and DSP? Copy the table of used resources to your lab report.

Analysis:

The Zynq device 7000 Family has the following features:

- Based on Xilinx System on Chip Architecture
- Dual-core or single-core ARM Cortex-A9 MPCores.
- Unified Level 2 cache (512 KB).
- Dual-ported, on-chip RAM (256 KB).
- 8-channel DMA.
- Interrupts and Timers: 1 GIC, 3 WDT, 2 Triple timer or counters.
- CoreSight debug and trace support for Cortex-A9.
- The range of devices in the Zynq-7000 family allows designers to target cost-sensitive as well as high-performance applications, as a result this can be used in many applications such as:
    1. Automotive driver assistance, driver information, and infotainment
    2. Broadcast camera
    3. Industrial motor control, industrial networking, and machine vision
    4. IP and Smart camera
    5. LTE radio and baseband
    6. Medical diagnostics and imaging
    7. Multifunction printers
    8. Video and night asvision equipment

Slice LUTs: Slice LUTs contains a set of flip-flops and multiplexers which store a predefined list of outputs for every combination of inputs and provide a faster way to retrieve the output of a logic operation.

Slice Register: Slice 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 updated to match the internally stored data.

Slice Logic:

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	679	0	53200	1.28
LUT as Logic	530	0	53200	1
LUT as Memory	149	0	17400	0.86
LUT as Distributed RAM	0	0		
LUT as Shift Register	149	0		
Slice Registers	548	0	106400	0.52
Register as Flip Flop	548	0	106400	0.52
Register as Latch	0	0	106400	0
F7 Muxes	0	0	26600	0
F8 Muxes	0	0	13300	0

Table 4: Slice Logic table

Block RAM: Block RAM, or block memory, is RAM that is embedded throughout the FPGA for storing data.

Site Type	Used	Fixed	Available	Util%
Block RAM Tile	0.5	0	140	0.36
RAMB36/FIFO*	0	0	140	0
RAMB18	1	0	280	0.36
RAMB18E1 only	1			

Table 5: Block RAM table

DSP: Xilinx FPGAs and SoCs are ideal for high-performance or multi-channel digital signal processing (DSP) applications that can take advantage of hardware parallelism. The DSP slices enhance the speed and efficiency of many applications beyond digital signal processing, such as wide dynamic bus shifters, memory address generators, wide bus multiplexers, and memory-mapped I/O registers.

Site Type	Used	Fixed	Available	Util%
DSPs	0	0	220	0

Table 6: DSP utilization table