Digital System Design SS2024

Lab 2: Parallel Processing (Digital Audio Filter)

Report submitted by:

Soumya Ranjan Sabat
Matrikelnr: 1127993

Azaz Hassan Khan
Matrikelnr: 1128032

June/07/2024

Content

1 Introduction 3
1.1 Tasks 3
1.1.1 I2S data format for audio data exchange 3
1.1.2 Analysis and understanding RTL Design 3
1.1.3 Top Level Inputs and Outputs 4
1.1.4 Interfacing Seven Segment Display 5
1.1.5 Synthesis and Analysis of Utilization report 6
2 References 10

1 Introduction

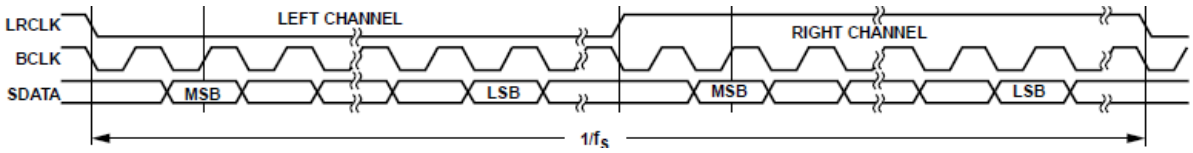
The main purpose of this lab is to display selected filter number on the attached 7-segment display. The 7- segment display is attached to the Pmod JA1+JB1 of ZedBoard. Parallel processing in FPGAs is demonstrated by analysing the implementation of algorithms for Digital filters.

- Study of the top-level RTL design
- VHDL design of the Camera Emulator
- Simulation of the Camera Emulator
- Test of HDMI Display Controller on ZedBoard

1.1 Tasks

1.1.1 I2S data format for audio data exchange

Q1. 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!



- LRCLK: Serial Data Port Frame Clock
- BCLK: Serial Data Port Bit Clock
- SDATA: Audio serial data

Serial data port Bit clock, BCLK has a frequency configured in accordance with the desired sampling bit rate. On every Negative edge trigger of BCLK one audio data bit is transmitted. The whole frame is sent twice, one to the left channel and one to the right channel of the Zync DSP block. LRCLK has duty cycle of 50% and on every negative edge of LRCLK data is sent to right channel and on every positive edge of LRCLK data is sent to left channel.

1.1.2 Analysis and understanding RTL Design

Q1. What is the purpose of the multiplexer hphone_l_i and hphone_r_i?

Input is being sampled by the audio codec from input GPIO pins. Sampled inputs are subjected to low pass filters to cancel the external noise associated with it. There are three filters with different lengths and filtering capabilities and one dsp block with no filtering. Any one of the three filters or the block with no filtering can be selected by the user. Hphone_l_i and Hphone_r_i takes the filtered outputs as input and user filter choice as selection bits and delivers corresponding output.

User selection for MUX output:

00	No Filtering
01	Filter with length 8
10	Filter with length 16
11	Filter with length 32

Q2. What is the function of Inst_dsp_block?

It takes AD converted signal from Audio Codec and sends the same to output without filtering.

Q3. Which port is connected to the serial data output of ADC?

ADC_SDATA/GPIO1 (pin 26): ADC Serial Output Data (ADC_SDATA).

Q4. Which port is connected to the serial data input of DAC?

DAC_SDATA/GPIO0 (pin 27): DAC Serial Input Data (DAC_SDATA).

Q5. Which clock frequency is used for signal MCLK?

Clock frequency of MCLK is 24MHz and sampling frequency of ADC and DAC is 48 KHz.

Q6. What is the filter length N of the three low pass filters?

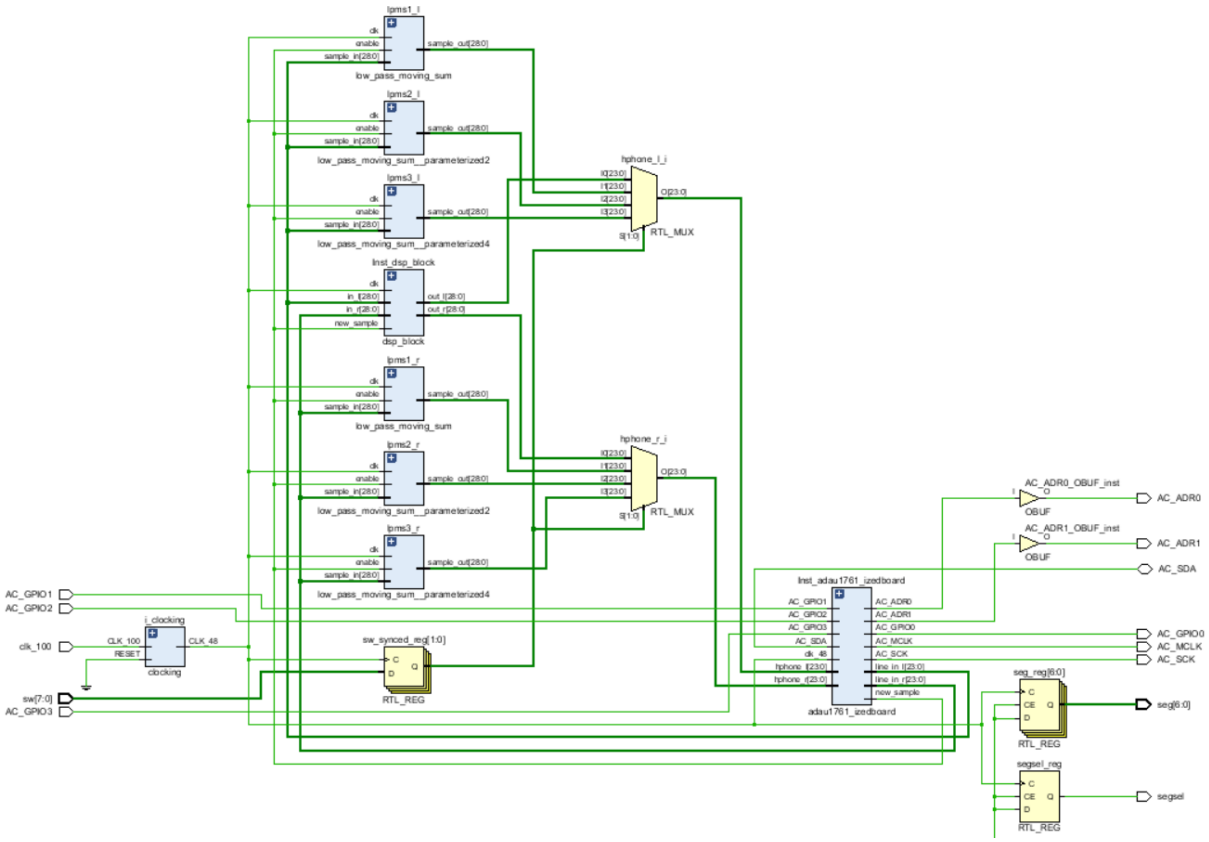
Filter lengths N of the three low pass filters are 8, 16 and 32 respectively.

Cutoff Frequency, Fc = sampling frequency/ filter length

For Filter lengths:

- N =8, Fc = 6 KHz
- N=16, Fc = 3 KHz
- N=32, Fc = 1.5 KHz

1.1.3 Top Level Inputs and Outputs



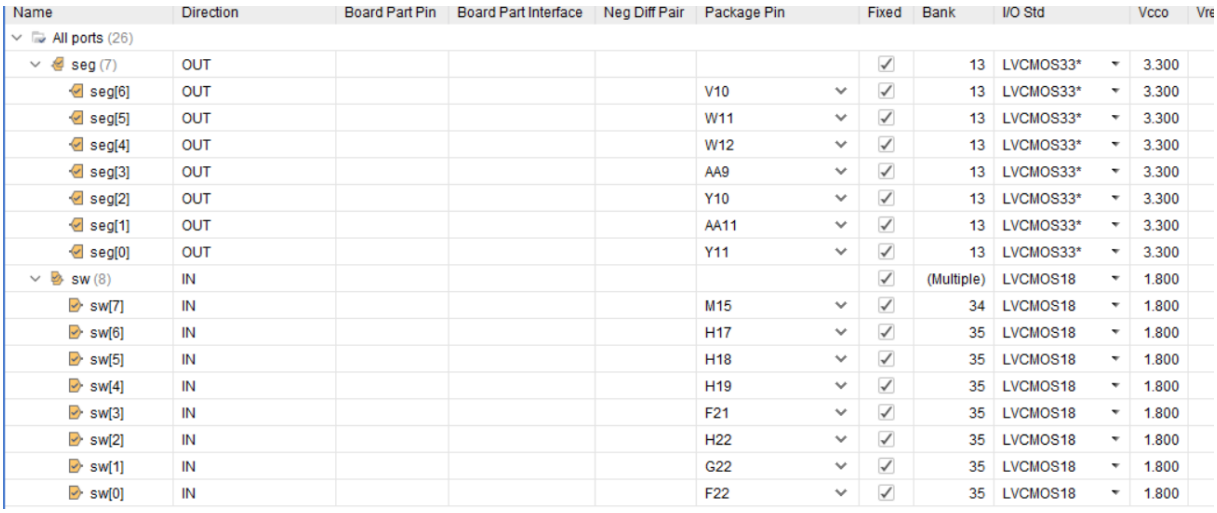
Input Signals	Description
AC_GPIO1	Digital Audio Serial Data ADC Output (I2S MOSI)
AC_GPIO2	Digital Audio Bit Clock Input/Output (I2S_bclk)
clk_100	Clock configuration
sw [7:0]	filter select
AC_GPIO3	Digital Audio Left-Right Clock Input/Output (I2S_LR)

Output Signals	Description
AC_ADR0	I2C Address Bit 0/SPI Latch Signal
AC_ADR1	I2C Address Bit 1/SPI Data Input
AC_SDA	I2C Serial Data Interface(bidirectional)
AC_GPIO0	Digital Audio Serial-Data DAC Input (I2S MISO)
AC_MCLK	Master Clock input(codec)
AC_SCK	I2C Serial Data interface
seg [6:0]	7 segments
segsel	cathode

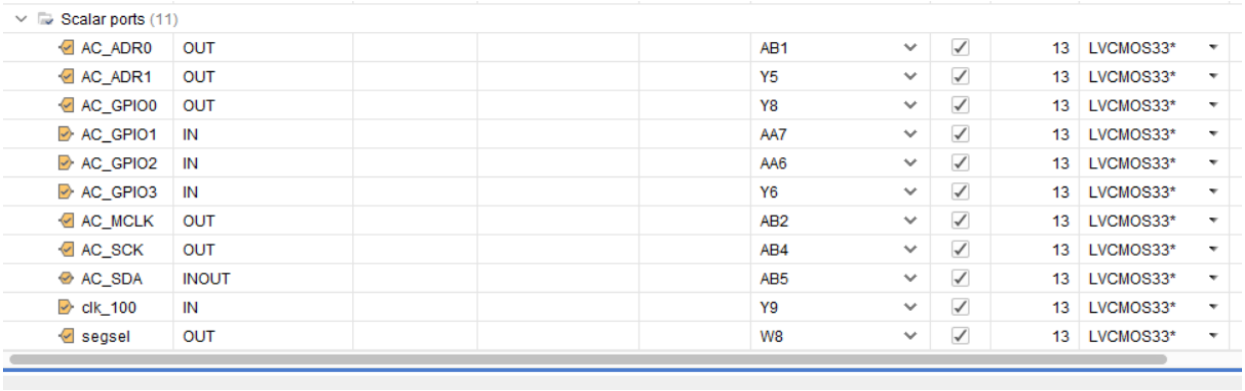
1.1.4 Interfacing Seven Segment Display

Q1. To display selected filter number on the 7-segment display, this is attached to the Pmod JA1+JB1 of ZedBoard and to add IO port settings for communication with pmod.

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



Name	Direction	Board Part Pin	Board Part Interface	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vre
All ports (26)										
seg (7)	OUT					☑	13	LVCMOS33*	3.300	
seg[6]	OUT				V10	☑	13	LVCMOS33*	3.300	
seg[5]	OUT				W11	☑	13	LVCMOS33*	3.300	
seg[4]	OUT				W12	☑	13	LVCMOS33*	3.300	
seg[3]	OUT				AA9	☑	13	LVCMOS33*	3.300	
seg[2]	OUT				Y10	☑	13	LVCMOS33*	3.300	
seg[1]	OUT				AA11	☑	13	LVCMOS33*	3.300	
seg[0]	OUT				Y11	☑	13	LVCMOS33*	3.300	
sw (8)	IN					☑	(Multiple)	LVCMOS18	1.800	
sw[7]	IN				M15	☑	34	LVCMOS18	1.800	
sw[6]	IN				H17	☑	35	LVCMOS18	1.800	
sw[5]	IN				H18	☑	35	LVCMOS18	1.800	
sw[4]	IN				H19	☑	35	LVCMOS18	1.800	
sw[3]	IN				F21	☑	35	LVCMOS18	1.800	
sw[2]	IN				H22	☑	35	LVCMOS18	1.800	
sw[1]	IN				G22	☑	35	LVCMOS18	1.800	
sw[0]	IN				F22	☑	35	LVCMOS18	1.800	



Scalar ports (11)								
AC_ADR0	OUT				AB1	✓	13	LVCMOS33*
AC_ADR1	OUT				Y5	✓	13	LVCMOS33*
AC_GPIO0	OUT				Y8	✓	13	LVCMOS33*
AC_GPIO1	IN				AA7	✓	13	LVCMOS33*
AC_GPIO2	IN				AA6	✓	13	LVCMOS33*
AC_GPIO3	IN				Y6	✓	13	LVCMOS33*
AC_MCLK	OUT				AB2	✓	13	LVCMOS33*
AC_SCK	OUT				AB4	✓	13	LVCMOS33*
AC_SDA	INOUT				AB5	✓	13	LVCMOS33*
clk_100	IN				Y9	✓	13	LVCMOS33*
segsel	OUT				W8	✓	13	LVCMOS33*

```
begin
process(clk_48)
   begin
   if rising_edge(clk_48) then
      sw_synced <= sw;
      ------------------------make your changes here
      segsel<='0'; -- 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<="1111111";
      END CASE;
   ------------------------
   end if;
end process;
```

1.1.5 Synthesis and Analysis of Utilization report

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

1.Slice LUT's: LUT's defined as Look Up Tables store predefined outputs for every combination of inputs. LUT in FPGA is a collection of hard-wired logic gates. Slice LUT is a collection of such LUT's.

1. Slice Logic

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	677	0	53200	1.27
LUT as Logic	528	0	53200	0.99
LUT as Memory	149	0	17400	0.86
LUT as Distributed RAM	0	0		
LUT as Shift Register	149	0		
Slice Registers	547	0	106400	0.51
Register as Flip Flop	547	0	106400	0.51
Register as Latch	0	0	106400	0.00
F7 Muxes	0	0	26600	0.00
F8 Muxes	0	0	13300	0.00

2.Slice Registers: Slice Registers are group of flip-flops which store the bit pattern. For every clock cycle the input data is latched and is stored internally and the output data is updated to match the internally stored data. Utilization of Slice LUT's and Slice Registers is shown in the picture below.

3.Block RAM: FPGA contains two types of internal RAM's. Block RAM and Distributed RAM. Block RAM serves as a relatively larger memory structure. They are used to store a bunch of data on chip for example synthesizing memory and FIFO functions. RAM utilization is shown in the picture below.

2. Memory

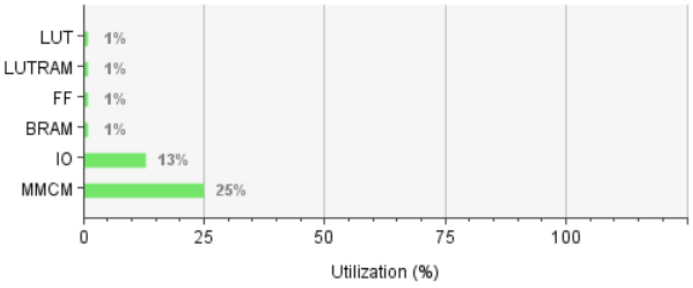
Site Type	Used	Fixed	Available	Util%
Block RAM Tile	0.5	0	140	0.36
RAMB36/FIFO*	0	0	140	0.00
RAMB18	1	0	280	0.36
RAMB18E1 only	1			

4.DSP: These blocks are used for DSP applications; they can be used to implement various complex calculations at a faster rate. They are highly optimized for maximum performance and minimum resource utilization. DSP Block utilization is shown below.

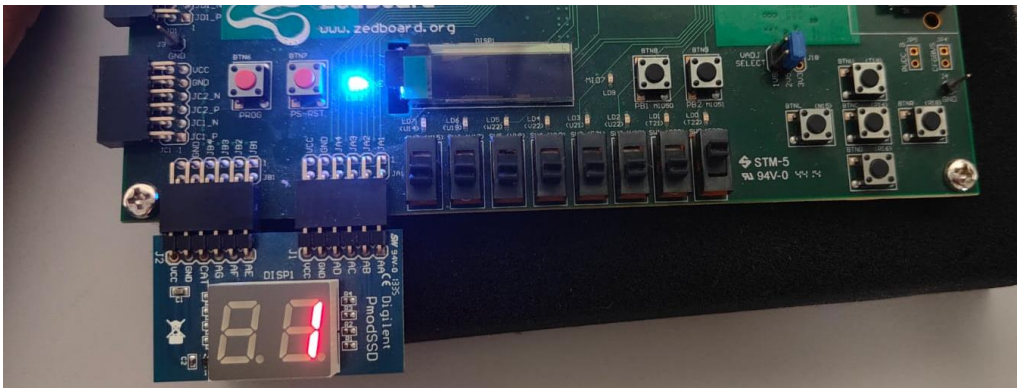
3. DSP

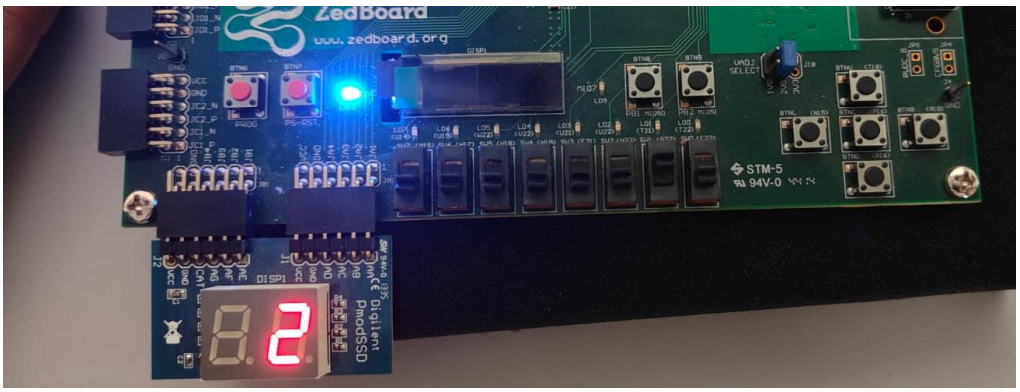
Site Type	Used	Fixed	Available	Util%
DSPs	0	0	220	0.00

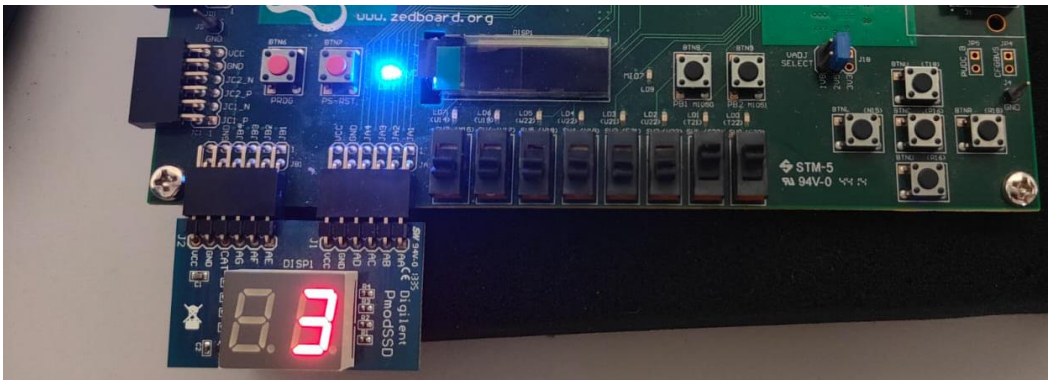
Resource	Utilization	Available	Utilization %
LUT	677	53200	1.27
LUTRAM	149	17400	0.86
FF	547	106400	0.51
BRAM	0.50	140	0.36
IO	26	200	13.00
MMCM	1	4	25.00

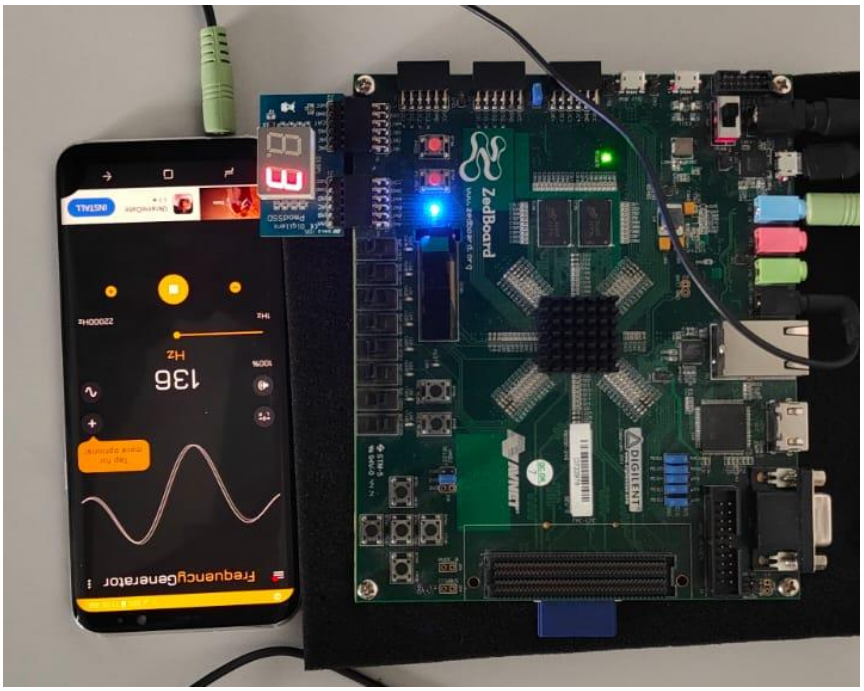


1.1.6 Output & Results






www.zedboard.org
DISP1
STM-5
94V-0
DISP1
Digilent
PmodSSD


ZedBoard
136
Hz
FrequencyGenerator

2 References

[1] "VHDL Programming by Example" by Douglas L. Perry

[2] "VHDL: Analysis and Modeling of Digital Systems" by Zainalabedin Navabi

[3] "FPGA Prototyping by VHDL Examples: Xilinx MicroBlaze MCS SoC" by Pong P. Chu

[4] "Digilent PmodSSD Peripherall Modulle Board Refference Manual"

[5] "Zynq Evaluation and Development) Hardware User’s Guide"