|  |
| --- |
| 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](#_Toc168626282)

[1.1 Tasks 3](#_Toc168626283)

[1.1.1 I2S data format for audio data exchange 3](#_Toc168626284)

[1.1.2 Analysis and understanding RTL Design 3](#_Toc168626285)

[1.1.3 Top Level Inputs and Outputs 4](#_Toc168626286)

[1.1.4 Interfacing Seven Segment Display 5](#_Toc168626287)

[1.1.5 Synthesis and Analysis of Utilization report 6](#_Toc168626288)

[2 References 10](#_Toc168626289)

# Introduction

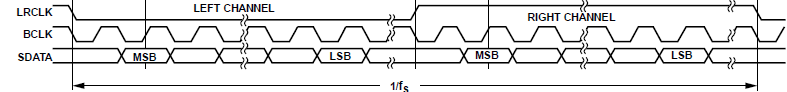
The main purpose of this lab is to display selected filter number on the attached 7-segment display. The 7- segment display is attached to the Pmod JA1+JB1 of ZedBoard. Parallel processing in FPGAs is demonstrated by analysing the implementation of algorithms for Digital filters.

* 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

## Tasks

### 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.

### 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?

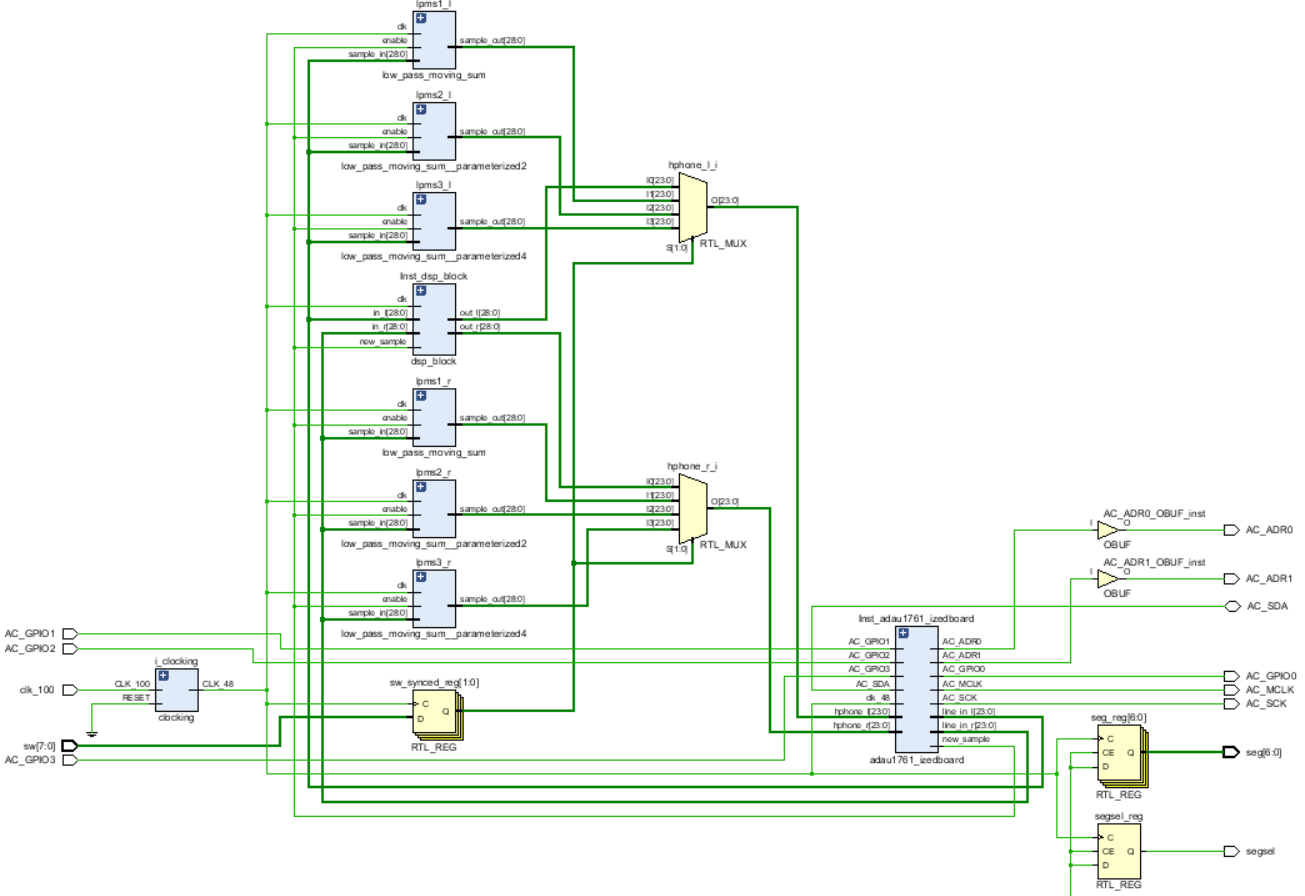
Filter lengths N of the three low pass filters are 8, 16 and 32 respectively.

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

### Top Level Inputs and Outputs



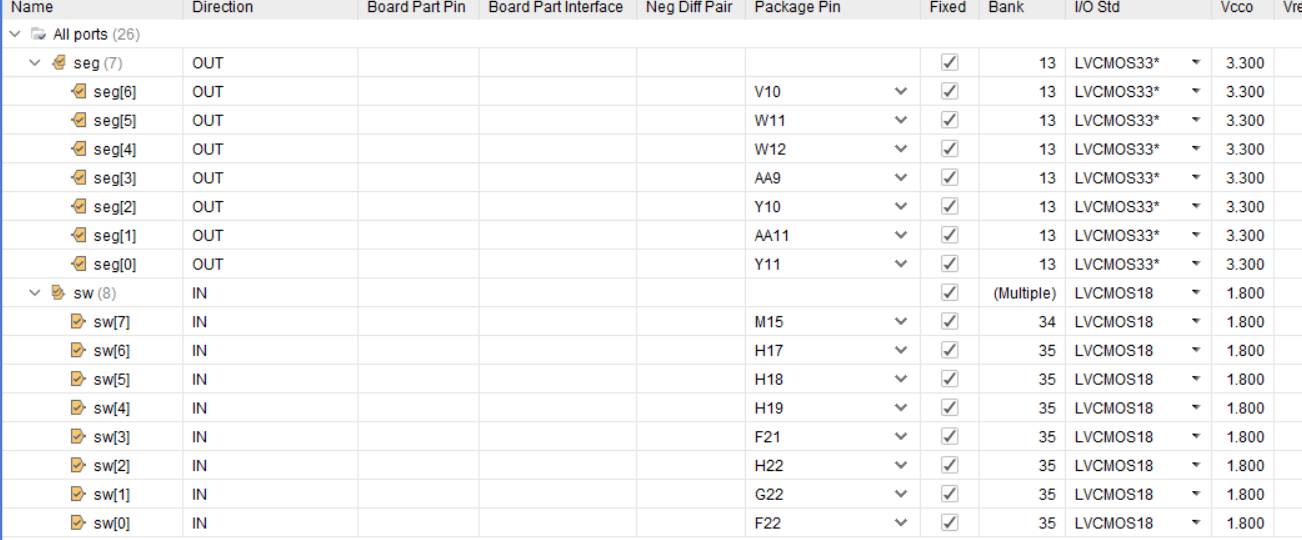
|  |  |
| --- | --- |
| **Input Signals** | **Description** |
| AC\_GPIO1 | Digital Audio Serial Data ADC Output (I2S MOSI) |
| AC\_GPIO2 | Digital Audio Bit Clock Input/Output (I2S\_bclk) |
| clk\_100 | Clock configuration |
| sw [7:0] | filter select |
| AC\_GPIO3 | Digital Audio Left-Right Clock Input/Output (I2S\_LR) |

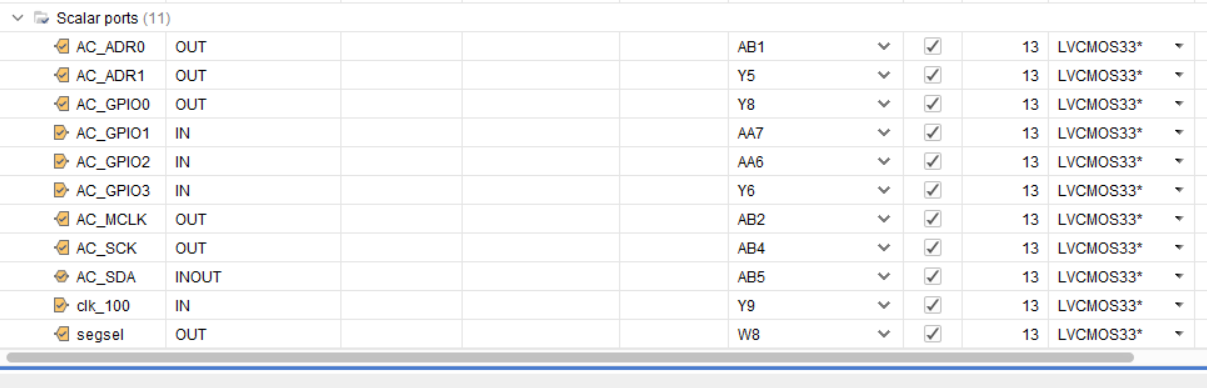
|  |  |
| --- | --- |
| **Output Signals** | **Description** |
| AC\_ADR0 | I2C Address Bit 0/SPI Latch Signal |
| AC\_ADR1 | I2C Address Bit 1/SPI Data Input |
| AC\_SDA | I2C Serial Data Interface(bidirectional) |
| AC\_GPIO0 | Digital Audio Serial-Data DAC Input (I2S MISO) |
| AC\_MCLK | Master Clock input(codec) |
| AC\_SCK | I2C Serial Data interface |
| seg [6:0] | 7 segments |
| segsel | cathode |

### 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 |



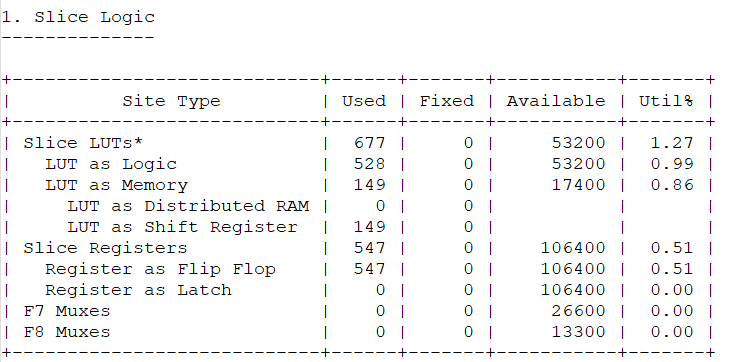


|  |
| --- |
| begin  process(clk\_48)  begin  if rising\_edge(clk\_48) then  sw\_synced <= sw;  ------------------------make your changes here  segsel<='0'; -- define the segment (cathode)  case sw is  WHEN "00000000"=>seg<="0111111"; -- control 7-Segment  WHEN "00000001"=>seg<="0000110";  WHEN "00000010"=>seg<="1011011";  WHEN "00000011"=>seg<="1001111";  WHEN OTHERS=>seg<="1111111";  END CASE;  -------------------------  end if;  end process; |

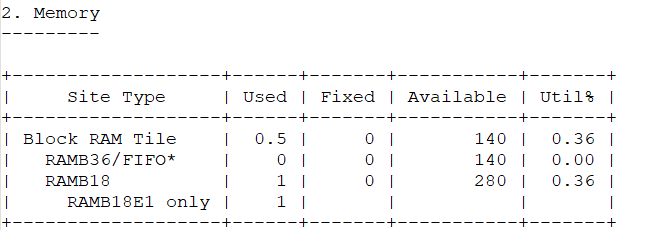
### Synthesis and Analysis of Utilization report

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

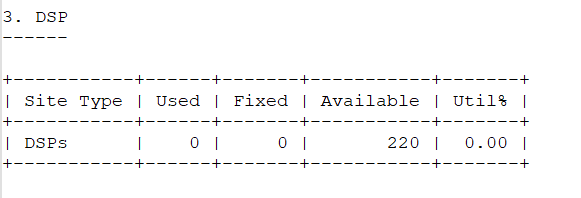
**1.Slice LUT’s:** LUT’s defined as Look Up Tables store predefined outputs for every combination of inputs. LUT in FPGA is a collection of hard-wired logic gates. Slice LUT is a collection of such LUT’s.

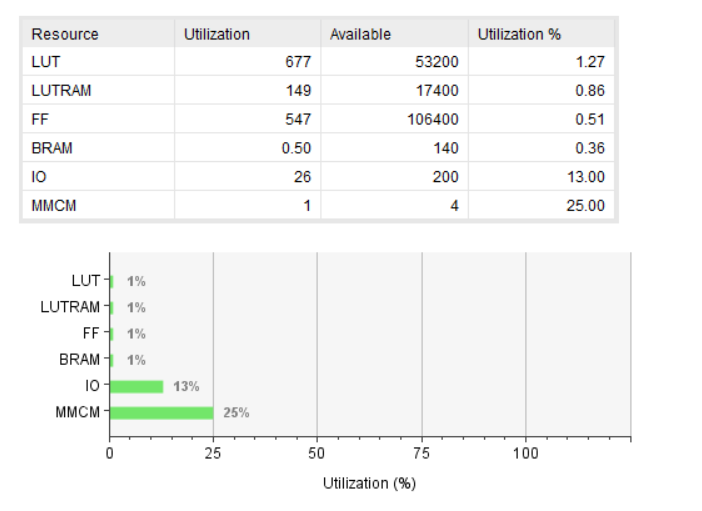


**2.Slice Registers:** Slice Registers are group of flip-flops which store the bit pattern. For every clock cycle the input data is latched and is stored internally and the output data is updated to match the internally stored data. Utilization of Slice LUT’s and Slice Registers is shown in the picture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

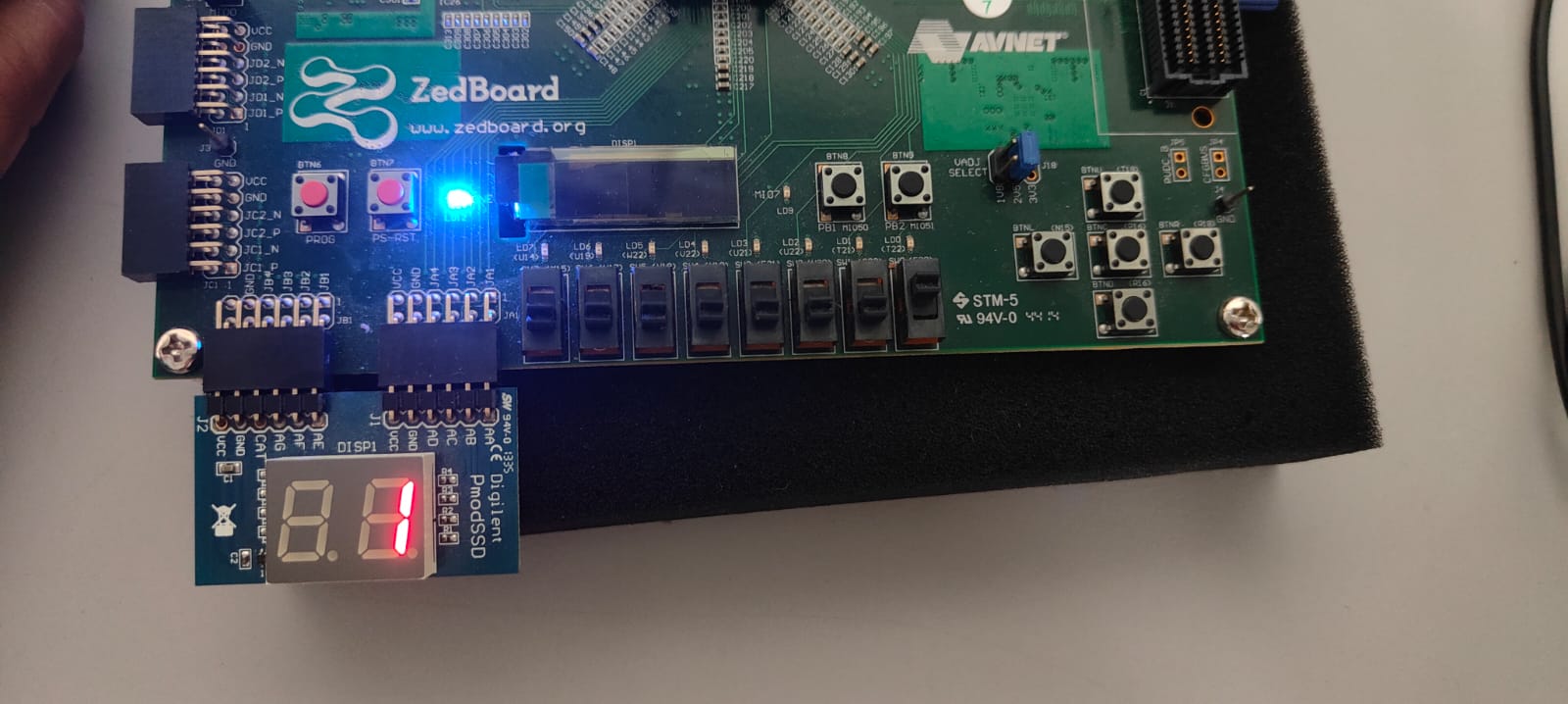


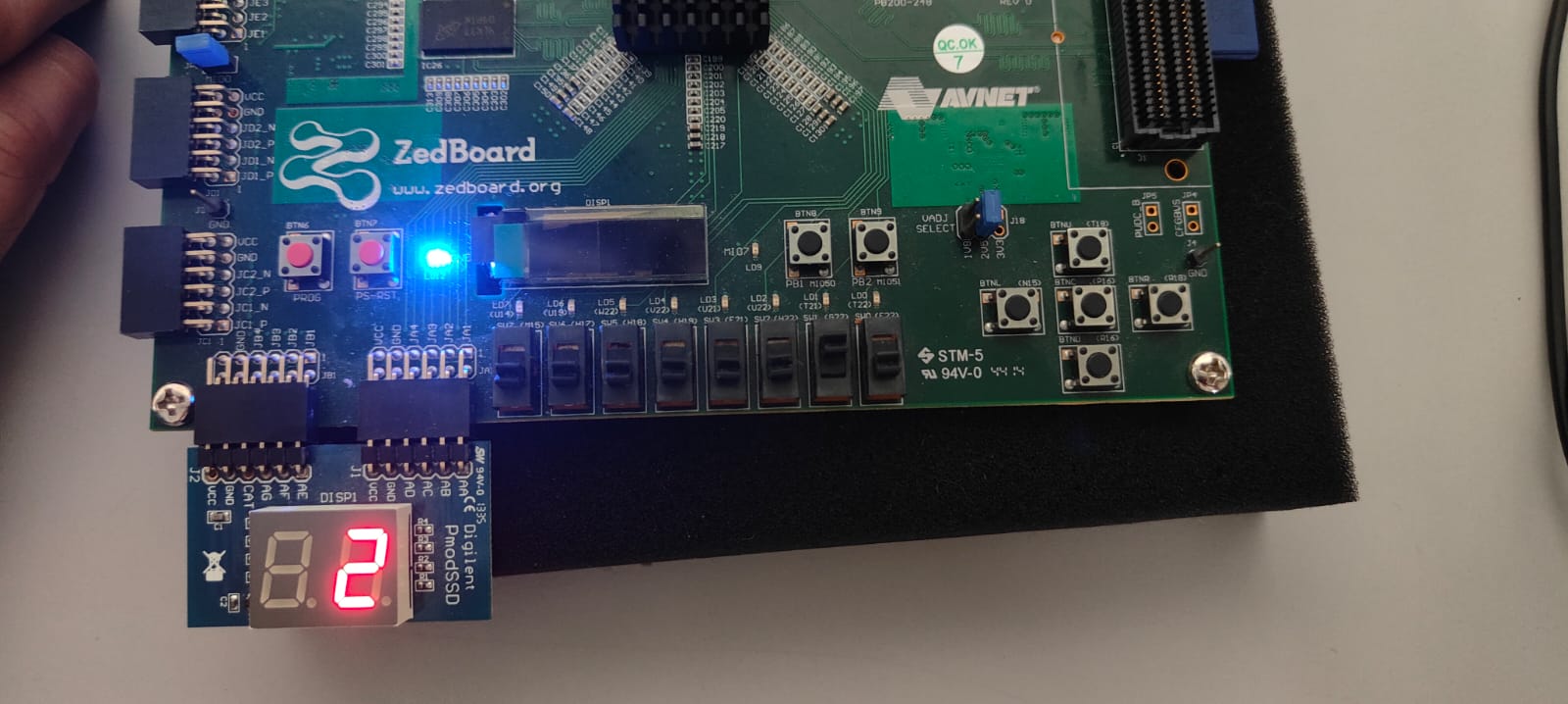
**4.DSP:** These blocks are used for DSP applications; they can be used to implement various complex calculations at a faster rate. They are highly optimized for maximum performance and minimum resource utilization. DSP Block utilization is shown below.

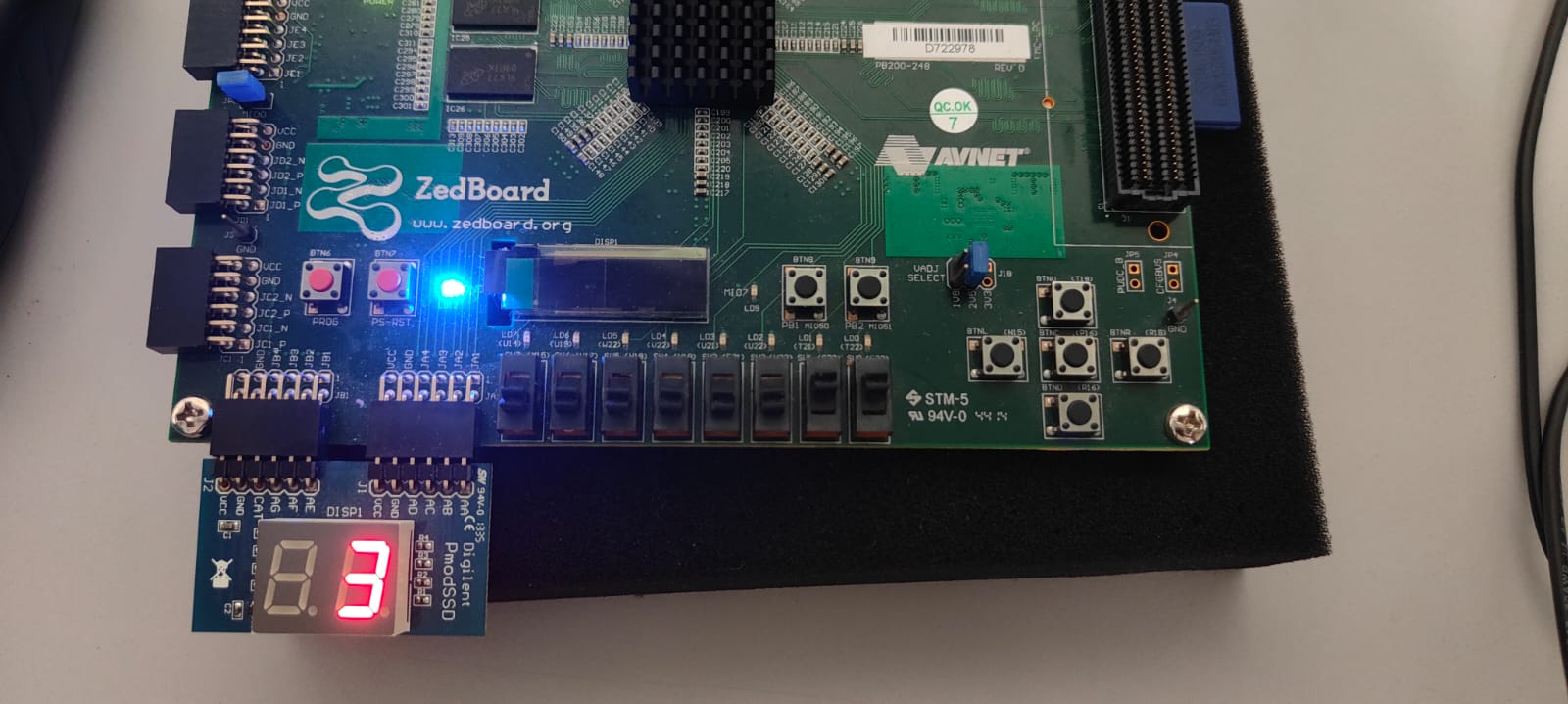


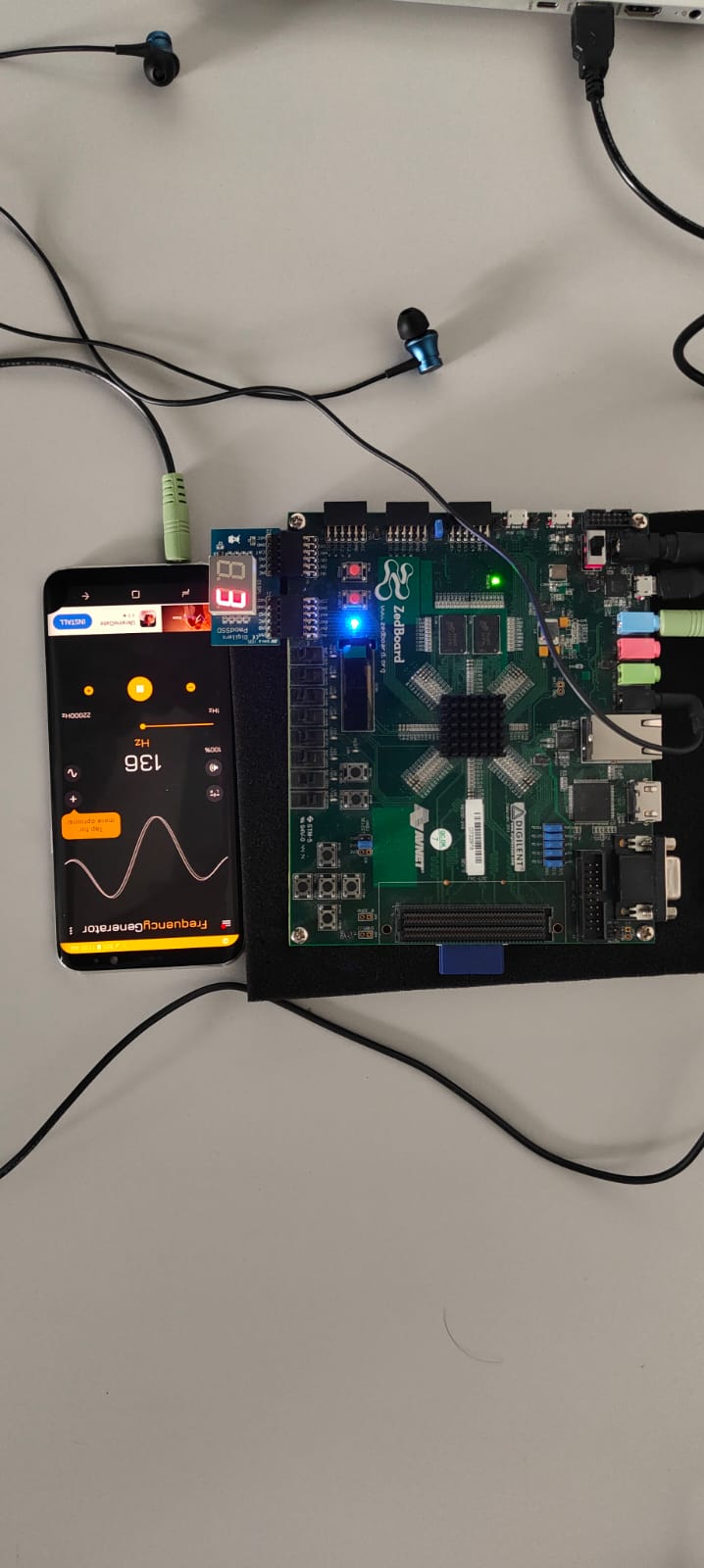


### Output & Results









# 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" |