AHMET FARUK ÇOLAK-22102104

Wearable Drum Kit

YouTube Link: https://youtu.be/SJlOc3SYOMo

1 Introduction

This project aims to create a wearable drum kit integrated into a backpack. Users can tap on designated switches placed on the backpack straps to trigger drum sounds played through headphones. This innovative approach aimed to make it possible to create rhythms on the go. The BASYS 3 FPGA board serves as the brain of the operation, detecting switch presses and generating corresponding digital signals for various drum sounds. These signals are then outputted to the user's headphones.

The inspiration for this project stemmed from the realization that rhythmic tapping on backpack straps could be translated into a functional musical instrument. Driven by a passion for music, this project aims to explore the intersection of music and technology.

Components Used: BASYS 3 FPGA board for processing the drum pad signals and microswitches as drum pads, empty PCB, some cables and headphones.

2 Methodology

Switch Placement: Switches on the backpack straps serve as drum pads, each producing a different drum sound. Switches are soldered onto the PCBs and they are on the backpack straps.

Signal Processing: The BASYS 3 board detects when a switch is pressed and sends out a digital signal for the sound.

Sound Generation: The internal 100MHz clock of the BASYS 3 FPGA board was utilized to generate the audio signals for the drum sounds. By manipulating the clock signal within the FPGA, distinct waveforms with varying frequencies and periods were produced, corresponding to different drum sounds. While the specific frequency and period of each waveform weren’t critical, ensuring harmony between the generated sounds was paramount for a cohesive audio experience.

Audio Output: The digitally generated waveforms are then transmitted through designated pins on the BASYS 3 board, enabling the audio signals to be directly outputted to headphones for user audibility.

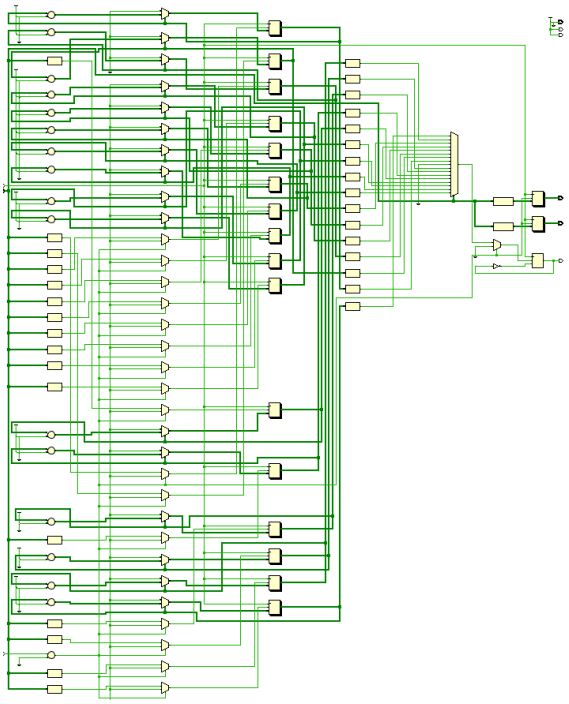

Figure.1 –Elaborated design of the system

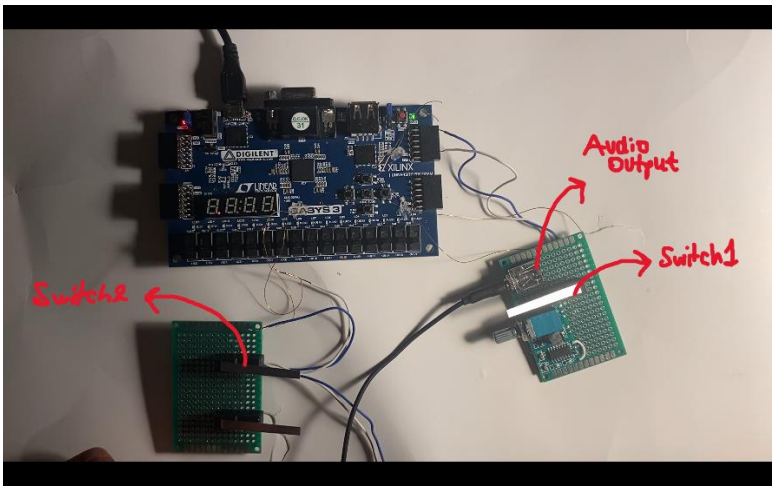

Figure.2 –Tecnical overview of the system

3 Result

The completed wearable drum kit backpack functions as intended. Users can play basic rhythms by tapping on the designated switch pads, with corresponding sounds heard through the headphones. The system provides a responsive and enjoyable drumming experience.

The project successfully translated the initial concept into a tangible prototype, allowing users to create rhythms while on the move. This aligns with the growing trend of wearable technology and exemplifies the fusion of art and engineering. However, due to the limited capabilities of the BASYS 3 board, the generated drum sounds don't fully replicate the complexity and richness of a real drum kit. Despite this, the implementation can be considered successful in demonstrating the feasibility of a wearable drum kit.

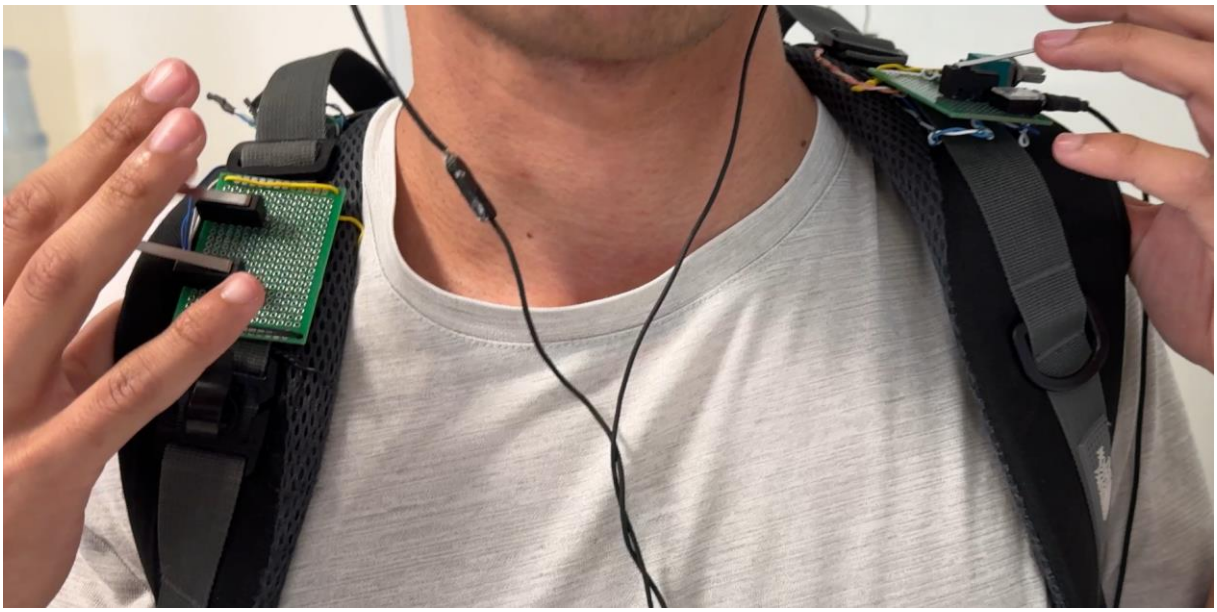

Figure.3 – Implementation of System

4 Conclusion

In conclusion, this project provided valuable insight into clock division techniques on the BASYS3 board. Through practical experimentation and theoretical exploration, I gained a deeper understanding of how to effectively utilize these components in digital circuit design.

This project served as a practical demonstration of how a wearable musical instrument can be designed and implemented. The process involved a comprehensive understanding of VHDL programming language, material selection, hardware assembly, and presentation skills – mirroring the end-to-end nature of engineering projects.

Appendix

5 Main Function

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;


entity main is
  Port ( ClkIn : in STD_LOGIC;
         switchs : in STD_LOGIC_VECTOR (15 downto 0);
         stop : in STD_LOGIC;
         frequ : out STD_LOGIC;
         gain : out STD_LOGIC;



end main;

architecture Behavioral of drum_kit is

  signal count15, count14, count2 : integer := 0;
  signal output : std_logic := '1';
  signal counter : integer := 0;

begin



  process (clkin) begin
  if stop = '0' then --makes it so that nothing can occur when the stop button is pushed
     if rising_edge(clkin) then --everything runs off of the rising edge of a 100 MHz clock (the system
clock)
```

```
    case (switchs) is
    when "1000000000000000" =>
          count15 <= count15 + 1;
          if count15 = 190000 then
             output <= not output;
             count15 <= 0;
          end if;

    when "0100000000000000" =>

             count14 <= count14 + 1;
             if count14 = 180000 then
                output <= not output;
                count14 <= 0;
             end if;


    when "0000000000000100" =
          count2 <= count2 + 1;
          if count2 = 90000 then
             output <= not output;
             count2 <= 0;
          end if;


    end case;

    end if;

end if;
```

```
  frequ <= output;
  end process;

end Behavioral;
```

6 Constraint

```
set_property PACKAGE_PIN W5 [get_ports ClkIn]
  set_property IOSTANDARD LVCMOS33 [get_ports ClkIn]
  create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports ClkIn]

set_property PACKAGE_PIN W19 [get_ports {switch[15]}]
set_property IOSTANDARD LVCMOS33 [get_ports {switch[15]}]

set_property PACKAGE_PIN T1 [get_ports {switch[14]}]
set_property IOSTANDARD LVCMOS33 [get_ports {switch[14]}]

set_property PACKAGE_PIN T17 [get_ports {switch[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {switch[2]}]

set_property PACKAGE_PIN U18 [get_ports {stop}]
set_property IOSTANDARD LVCMOS33 [get_ports {stop}]

set_property PACKAGE_PIN A14 [get_ports {frequ}]
set_property IOSTANDARD LVCMOS33 [get_ports {frequ}]
```

```
set_property PACKAGE_PIN A16 [get_ports {gain}]
set_property IOSTANDARD LVCMOS33 [get_ports {gain}]
```