IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026

DOI: 10.17148/IJIREEICE.2026.14206

# A Spirometer-like System, Providing Indication About Lung Health, Using Air flow Sensor, FPGAs and VHDL

**Dr Evangelos I. Dimitriadis[1], Leonidas Dimitriadis[2]**

Department of Computer, Informatics and Telecommunications Engineering, International Hellenic University, End of Magnisias Str, 62124 Serres Greece[1]

Undergraduate student, Department of Information and Electronic Engineering, International Hellenic University, 57400, Sindos Thessaloniki, Greece[2]

**Abstract:** A spirometer-like system, based on MD0550 air flow sensor, FPGAs and VHDL, is presented here. The system is capable of acting as an indicator for lung health by measuring forced expiratory air flow of a human, versus time and subsequently calculate and display both air flow values and the ratio FEV1/FVC of Forced Expiratory Volume in first second (FEV1) to Forced Vital Capacity (FVC). The above ratio is critical for providing useful indication about human lung health and its values must be equal or higher to 0.7 for a healthy person. Our system except from displaying air flow and ratio values, also uses a LED system for providing information about low, medium or high air flow values, with limits set by user, thus lighting on blue, green or red external LEDs, respectively. Half left or right of board LEDs also light up for low or high air flow values, respectively. In case that FEV1/FVC ratio values are below 0.7 then buzzer sounds and white external LED is ON, while yellow LED lights up in case that FEV1/FVC ratio values are equal or higher than the critical value of 0.7. The system uses DE10-Lite FPGA board, is cheap to manufacture, easy to use and can be also combined with IoT systems, allowing it to share valuable health information of a human, with its doctor.

**Keywords:** MD0550 air flow sensor, air flow monitoring, lung health, FEV1/FVC ratio, FPGA, VHDL, Buzzer, LEDs.

## I. INTRODUCTION

It is well known that FPGAs have attracted attention of researchers in the recent years, for industrial as well as for a variety of other applications. [1-13] FPGAs have the main advantage of combining software and hardware, thus enabling hardware programming for a series of applications. The most used languages for FPGAs' programing are VHDL and Verilog and VHDL is the one used in our work.

Although a lot of work has been done concerning implementation of FPGAs in a variety of applications as mentioned above, there are few works [14-17] dealing with spirometer applications using, some of them, FPGA boards. These works present an FPGA-based rapid wheezing detection system, or show experimental results for spirometry-on-chip. Others present Field Programmable Gate Array (FPGA) respiratory monitoring system using a flow microsensor and an accelerometer, or show FPGA-based error correction in MEMS sensors used for respiration monitoring system.

All of the above works do not solve the problem of presenting an FPGA-based, simple, portable, easy to use and low cost spirometer-like system, capable of providing an important indication about lung health, which is the main subject of our work.

We present here an air flow sensor-based system, which can monitor and display air flow values during human forced expiratory process and subsequently provide valid and also valuable information about calculated and displayed ratio FEV1/FVC of Forced Expiratory Volume in first second (FEV1) to Forced Vital Capacity (FVC). The above ratio plays a critical role in the diagnosis of human lung health. Our spirometer-like system is also equipped with a complete LED system, activating each corresponding LED for emphasizing the scales of air flow input values as well as FEV1/FVC ratio values. Buzzer system also sounds in case that ratio FEV1/FVC<0.7.

Another benefit of our system is that it can work with a variety of air flow sensors with different sensitivities, as long as they have an analog output.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026

DOI: 10.17148/IJIREEICE.2026.14206

## II. DESIGN OVERVIEW AND OPERATION OF THE SYSTEM

Figure1 presents device overview and operational units of our system, using FPGA DE10-Lite board, while Figure 2 presents circuit diagram of the system. Subsequent Figure 3 presents the implemented spirometer-like system of this work.

It is obvious from the above figures that our system, except from DE10-Lite FPGA board, contains also some basic circuit parts. First is air flow monitoring system, second is output LEDs system and third is buzzer alarm system. DE10-Lite FPGA board used here offers, its seven-segment displays for presenting input air flow sensor values, as well as FEV1/FVC ratio values and board LEDs mentioned above.

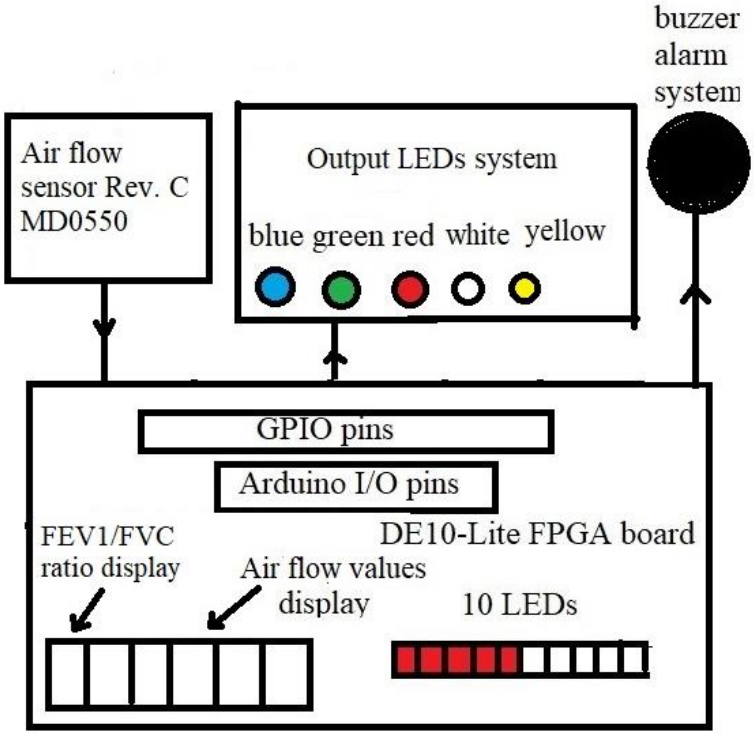



Figure 1: Device overview and operational units of our system.

Time is the other input value used here and it is provided by FPGA's clock.

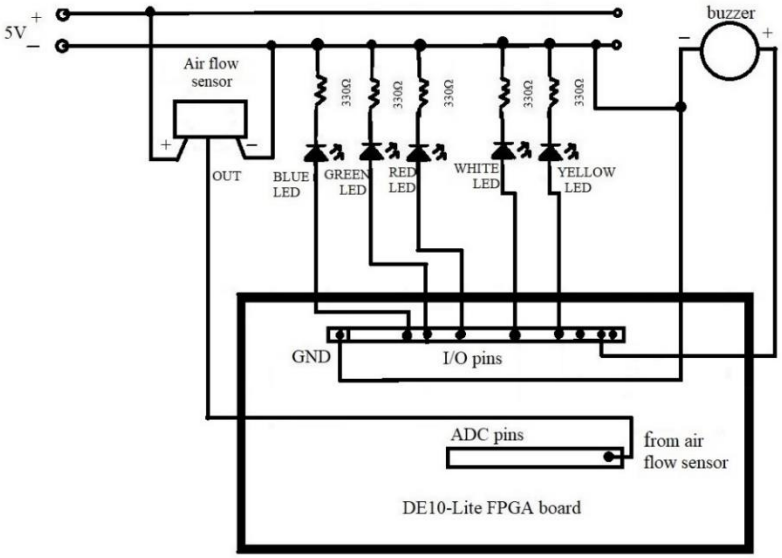



Figure 2: Circuit diagram of our system

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026

DOI: 10.17148/IJIREEICE.2026.14206

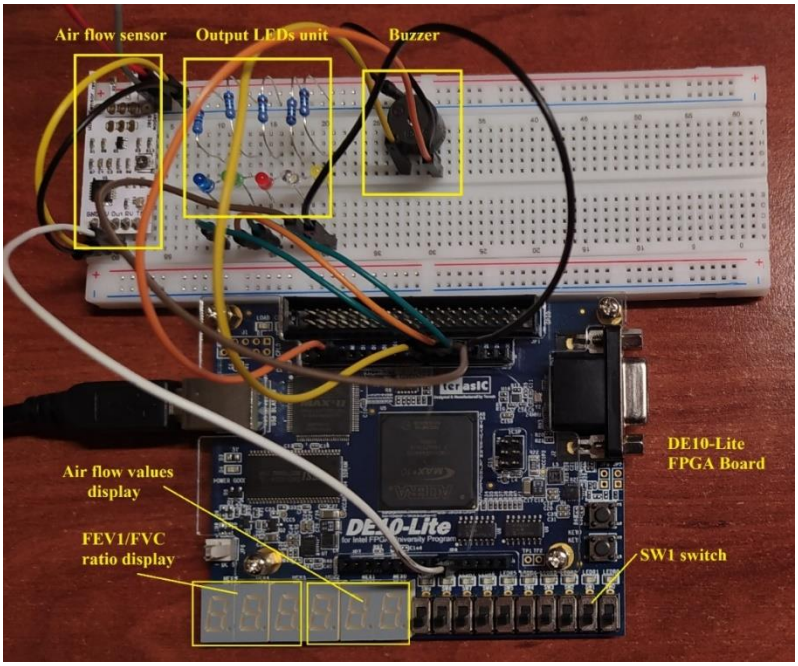



Figure 3: The implemented spirometer-like system of this work.

Our spirometer-like system starts operating as soon as power supply +5V is applied to the circuits and the VHDL program is sent via USB Blaster interface, to FPGA chip. Input values from both MD0550 air flow sensor unit and FPGA's clock are entered in our system. Analog to digital converter (ADC) of DE10-Lite proceeds to conversion and finally input voltage values are presented in seven-segment displays of the FPGA board. It must be mentioned here that air flow input voltage values (Vr) related to forced expiratory procedure, are divided into three basic zones shown below in Table1. The zones' limits are arbitrary chosen and they are related to MD0550 air flow sensor manual calibration and sensitivity. During calibration we paid attention to achieve a nearly zero input air flow value when no expiratory process is present. We also used healthy persons for expiratory process, in order to distinguish upper air flow input values for MD0550 sensor which reached 1.8V. During calibration process, persons with asthma were not able to reach the critical value of 0.7 for FEV1/FVC ratio, using our spirometer-like system.

Table 1: Characterization of air flow input values

| Air flow input values Vr (V) | Characterization – LEDs ON |
|---|---|
| Vr < 0.25 | Low values-Blue LED ON |
| Vr >= 0.25 AND Vr < 0.40 | Medium values-Green LED ON |
| Vr >= 0.40 | High values-Red LED ON |

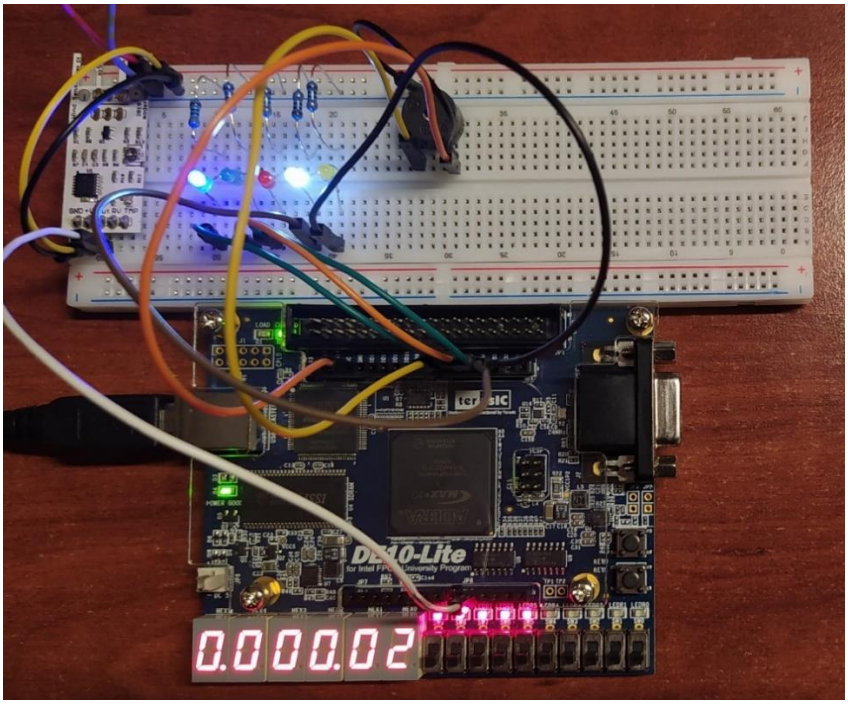
Upon our system's operation begins, simultaneously one additional to ADC conversion and displaying, basic process begins to execute, according to Table 1. Air flow input voltage values are checked and depending on their zone, corresponding external LED lights up, as presented in Table 1 and shown in Figures 4, 6, 7 and 8. Simultaneously to blue or red external LED lighting, left or right half of board LEDs light up, respectively, as shown in Figures 4, 7 and 8. We must mention that in Figure 4a, SW1 switch is OFF, thus the important procedure of calculating and displaying FEV1/FVC ratio values is not executed. Subsequently ratio value displayed is zero, which is lower than the critical value of 0.7, thus the system activates another basic procedure which lights up white external LED, in case that FEV1/FVC ratio values are lower than 0.7. In Figure 4b SW1 switch is ON and FEV1/FVC ratio values are displayed at left displays. As soon as SW1 switch goes to ON state the most important procedure of our system is activated and FEV1/FVC ratio values are calculated and displayed.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

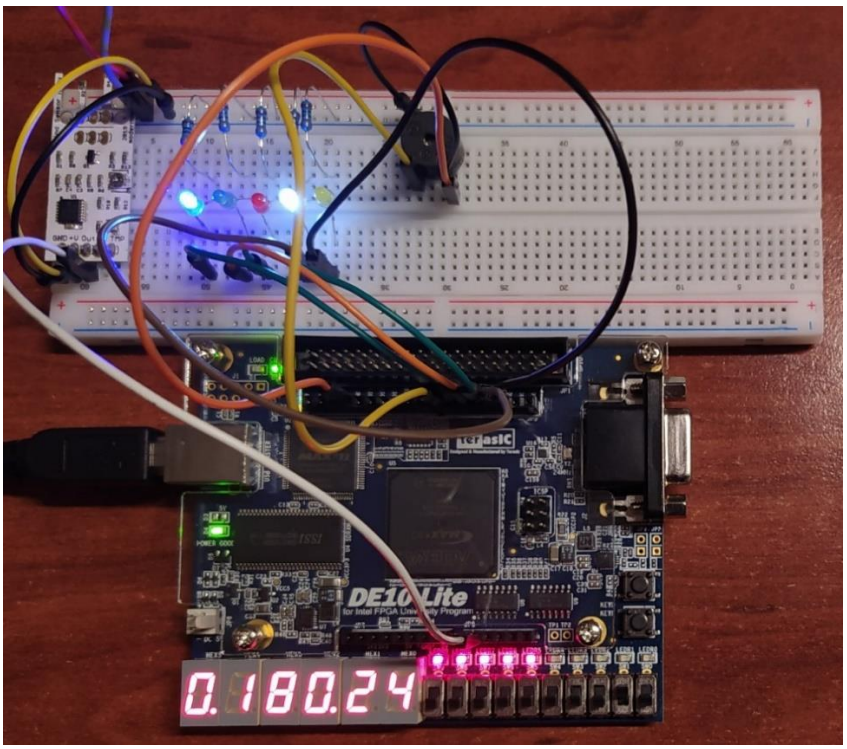
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026

DOI: 10.17148/IJIREEICE.2026.14206




a)




b)

Figure 4: a) System activated but SW1 switch is OFF. No ratio display. b) SW1 is ON and both air flow and ratio values are low, thus blue and white LEDs are ON and left half of board LEDs light up.

In case that FEV1/FVC ratio values are equal or higher than 0.7, yellow external LED is ON, indicating healthy lungs, as shown in Figure 8. We must also mention here that in case that SW1 switch is ON and FEV1/FVC ratio values are lower than 0.7, simultaneously with white LED activation our system turns buzzer to ON state.

The procedure that calculates FEV1/FVC ratio values is based upon Figure 5 which presents a normal forced expiratory volume–time graph [18]. The decisive role of the expiratory volume in the first second is clearly shown in Figure 5.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026

DOI: 10.17148/IJIREEICE.2026.14206

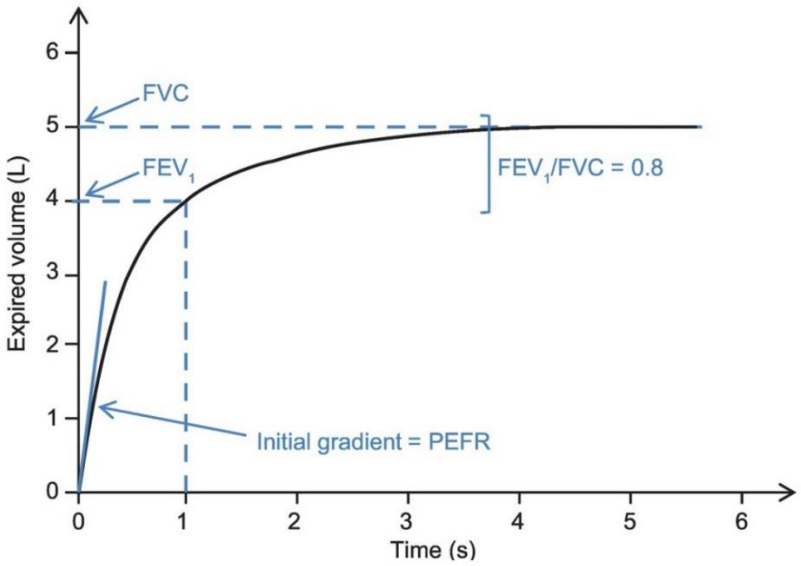



Figure 5: Normal forced expiratory volume–time graph.

According to Figure 5, our spirometer-like system is using four seconds time to complete the process of calculating FEV1/FVC ratio values and the process begins upon transition of SW1 board switch, from OFF to ON state.

Figure 6 presents our system with SW1 switch to ON state and simultaneously medium air flow input values and low FEV1/FVC ratio values (<0.7), thus green and white LEDs, respectively, are ON.

Figure 7 presents our system with SW1 switch to ON state and simultaneously high air flow input values and low FEV1/FVC ratio values (<0.7), thus red and white LEDs, respectively, are ON. Also right half of board LEDs are ON.

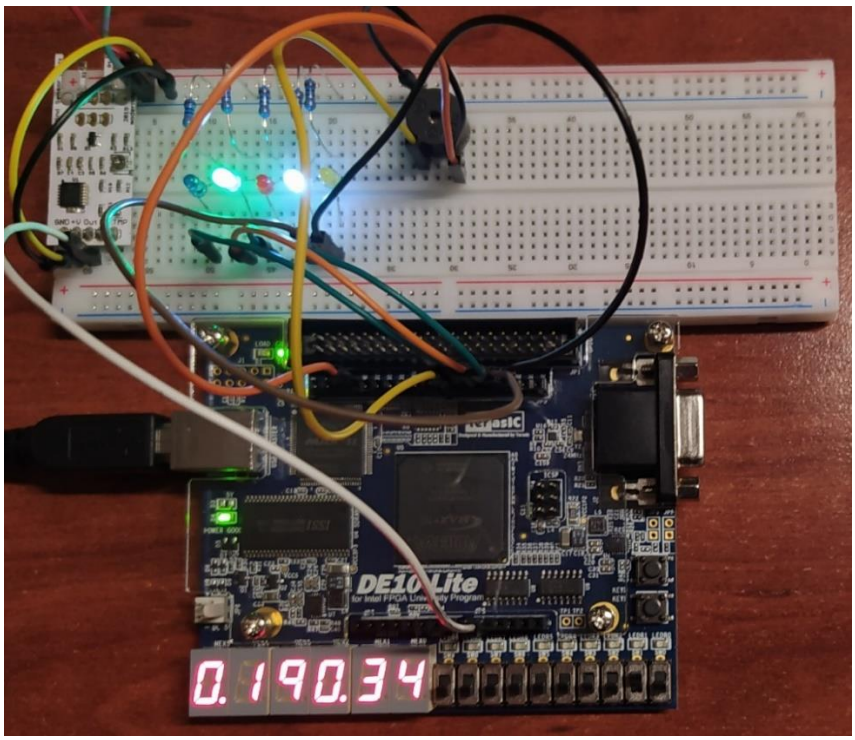



Figure 6: Medium air flow values and low ratio values, thus green and white LEDs are ON.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026

DOI: 10.17148/IJIREEICE.2026.14206

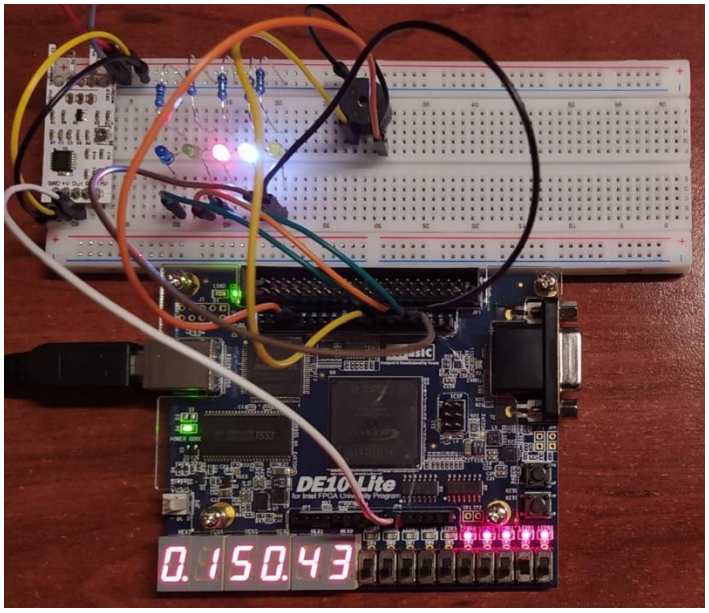



Figure 7: High air flow values and low ratio values, with red and white LEDs ON. Right half of board LEDs also ON.

Finally, Figure 8 presents our system with SW1 switch to ON state and simultaneously high air flow input values and high FEV1/FVC ratio values (>=0.7), thus red and yellow LEDs, respectively, are ON. Also right half of board LEDs are ON.

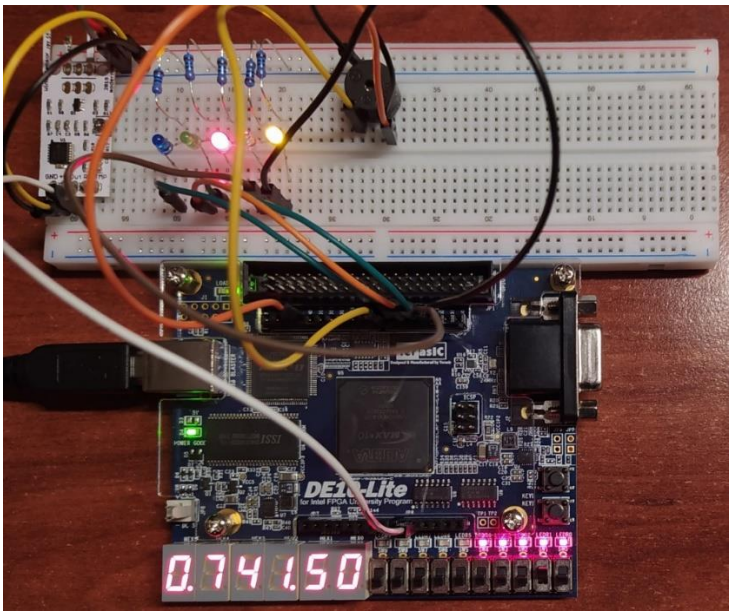



Figure 8: High air flow and ratio values, lighting up red and yellow LEDs. Right half of board LEDS are also ON.

It is clear from Figure 8 that healthy persons can achieve high air flow input values, which are prerequisites for achieving (FEV1/FVC) >=0.7 ratio values.

All of the above mentioned external LEDs are connected to I/O pins of the FPGA board and act as outputs for our system.

Another important fact of our system's design is that it receives input air flow voltage values periodically, ensuring continuous air flow change monitoring within four seconds critical time period, mentioned above.

Our spirometer-like system also offers the advantage of being easy to use. One can simply turn on the system, take a deep breath and start forced expiratory process towards MD0550 air flow sensor from 2-5 cm away, with simultaneous

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026
DOI: 10.17148/IJIREEICE.2026.14206

transition to ON state for SW1 switch. Testing our system with various participants, we noticed that forced expiratory process is equally effective with or without the use of a small straw, making our system even easier to use.

## III. PROGRAMING THE SYSTEM

We used Quartus Prime Lite Edition 21.1.1 to create the VHDL programs of our system. It must be mentioned here that before proceeding with the VHDL programming of our system, we had to set a series of parameters controlling the operation of DE10-Lite FPGA's Analog to Digital Converter (ADC). This converter plays a very important role in the whole system operation, since it converts the analogue input voltage from MD0550 air flow sensor connected to FPGA board to digital values, acting as main input of the system. The files created by the above ADC parameters setting are imported into the final project of our system.

A flowchart diagram, presenting main functions of our system is presented in Figure 9, while the APPENDIX contains the whole VHDL program.
It is clear that the system basically operates five functions. All of them use processes in VHDL programming language. These processes are running as long as the system is ON. Additionally SW1 switch of the FPGA board must be ON (logic 1), in order to activate the most important function (second from left in Figure 9), which calculates FEV1/FVC ratio and presents the results at first three seven-segment displays at the left of the board. All external or board's LEDs as well as buzzer system, are programmed to work as logic outputs, thus they are at the ON state if they receive binary 1 as logic output.

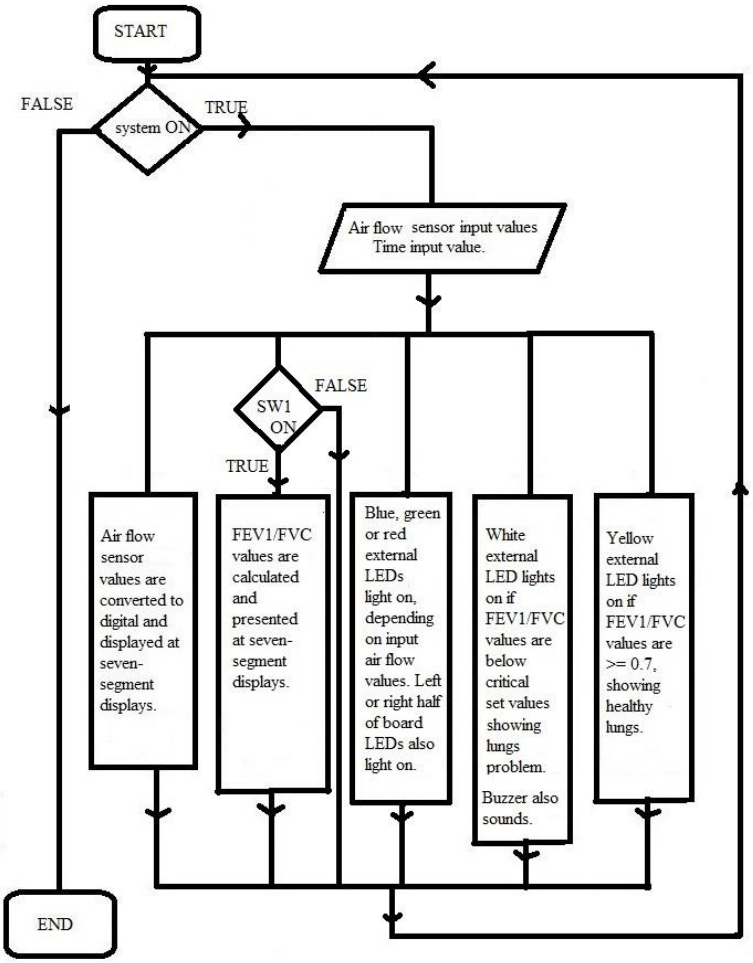



Figure 9: Flowchart diagram, presenting main functions-processes of our system.

First function of Figure 9 playing definitive role in system operation, uses analog input voltage values provided from MD0550 air flow sensor as main input, convert them to digital values and present the final result in three right seven-segment displays of the FPGA board. Calculated input sensor voltage values are used in other processes, in order to

DOI: 10.17148/IJIREEICE.2026.14206

activate external or internal LEDs. A separate process based on FPGA board's clock, is also activated upon system's transition to ON state. It is a very important process whenever time parameter is needed, but it is not shown as a separate function in Figure 9.

Third function shown in Figure 9 includes the process that activates blue LED and left half of board LEDs in case that input air flow values are in low zone. Green LED lights up for medium zone air flow input values, while red LED and right half of board LEDs light up if air flow input values are in high zone. Needless to say that our program gives the ability of setting air flow input values limits according to the sensitivity of air flow sensor, thus making the system capable of using a variety of sensors.

Functions four and five of Figure 9 are using the calculated from second function FEV1/FVC ratio, in order to activate external LEDs providing indication about lung health. Function four is checking if ratio (FEV1/FVC)<0.7 and in case that this holds true, white external LED and buzzer receive logic 1 and switch to operating mode. This fact indicates a problem in human lungs.

Finally fifth function of Figure 9 indicates healthy human lungs, in case that ratio (FEV1/FVC) >= 0.7, leading yellow external LED to ON state after receiving an output logic 1.

## IV. CONCLUSION

An FPGA-based, spirometer-like system is presented in this work. Our system is able to provide a precise indication about human lung health and has the main advantage of ease of use. MD0550 air flow sensor due to its great sensitivity, manages to perceive and convert, even a small air flow, into an analog signal, thus no other equipment is needed for expiratory procedure giving to our system the advantage of portability. Additionally user can blow air from 2-5 cm distance to sensor, with or without the use of a straw. Air flow input values as well as FEV1/FVC ratio values are shown in seven-segment displays of FPGA board. Our spirometer-like system also provides visualized information about the zone of input air flow values and FEV1/FVC ratio values, by activating corresponding external LEDs and FPGA board LEDs. Buzzer also sounds in case that FEV1/FVC ratio values are below the critical value of 0.7. The system can work with all commercial air flow sensors that provide an analog output and it is easy to be manufactured, providing also the benefit of low cost.

## ACKNOWLEDGEMENTS

We would like to express our warm thanks to **Theodora I. Dimitriadou, Eleni L. Fanara** and our students, for their participation in calibrating and testing our system.

## REFERENCES

[1]. M. Yussup, M.M. Ibrahim, L. Lombigit, N.A.A. Rahman and M.R.M. Zin, "Implementation of data acquisition interface using on-board field-programmable gate array (FPGA) universal serial bus (USB) link", Advanc. in Nuclear Research and Energy Development, AIP Conf. Proc. 1584, 69-72 (2014), doi: 10.1063/1.4866106

[2]. A. Gujar, International Journal of Computer Science and Information Technologies, "Image Encryption using AES Algorithm based on FPGA", vol 5, (5), 2014, 6853-6859

[3]. S. Singh, A.K. Saini, R. Saini, I.J. Image, Graphics and Signal Processing, "Interfacing the Analog Camera with FPGA Board for Real-time Video Acquisition" 2014, 4, 32-38, DOI: 10.5815/ijigsp.2014.04.04

[4]. S. Muthukrishnan and R. Priyadharsini, International Journal of Computer Science and Mobile Computing, "32-Bit RISC and DSP System Design in an FPGA" vol 3, issue 12, Dec. 2014, pg. 361-368

[5]. L. Chen, Y. Chang, L. Yan, IEEE Transactions on Geoscience and Remote Sensing, "On-orbit real-time variational image destriping: FPGA architecture and implementation", 10.1109/tgrs.2022.3140428, 2022, pp. 1-1

[6]. S. Yarlagadda, S. Kaza, A. Tummala, E. Babu, R. Prabhakar, Information Technology in Industry, "The reduction of Crosstalk in VLSI due to parallel bus structure using Data Compression Bus Encoding technique implemented on Artix 7 FPGA Architecture", 10.17762/itii.v9i1.151, 2021, Vol 9 (1), pp. 456-460

[7]. C. Du, Y. Yamaguchi, Electronics, "High-Level Synthesis Design for Stencil Computations on FPGA with High Bandwidth Memory", 2020, 9(8), 1275; https://doi.org/10.3390/electronics9081275

[8]. X. Hao, C. Lin and Q. Wu, Electronics, "A Parallel Timing Synchronization Structure in Real-Time High Transmission Capacity Wireless Communication Systems", 2020, 9(4), 652; https://doi.org/10.3390/electronics9040652

DOI: 10.17148/IJIREEICE.2026.14206

[9]. P. A. Bawiskar, R.K. Agrawal, International Journal of Innovative Research in Science, Engineering and Technology, "FPGA Based Home Security System" vol.4, issue 12, Dec. 2015, p. 12865-12869, DOI: 10.15680/IJIRSET.2015.0412139

[10]. K. Saroch, A. Sharma, IOSR Journal of Electronics and Communication Engineering, "FPGA Based System Login Security Lock Design Using Finite State Machine" vol 5, issue 3, Mar.-Apr. 2013, pp 70-75

[11]. R.S. Parikh, Int. Journal of Engineering Research and Application, "Alarm System Implementation on Field Programmable Gate Array" vol 8, issue 1, Jan. 2018, pp 01-04.

[12]. E. I. Dimitriadis and L. Dimitriadis, "A Simple, Low Cost and Multiple Input Alarm System, Functioning as Finite State Machine (FSM), Using VHDL and FPGAs", Journal of Active and Passive Electronic Devices, vol. 17, pp. 307–315, 2024.

[13]. E. I. Dimitriadis and L. Dimitriadis, "A g-sensor based alarm system, for multiple tilt sensor applications, using VHDL and FPGAs", Journal of Active and Passive Electronic Devices, vol. 18, pp. 119-129, 2024.

[14]. Bor-Shing Lin, Tian-Shiue Yen, "An FPGA-based rapid wheezing detection system", International Journal of Environmental Research and Public Health (IJERPH), 11(2), 1573-1593, February 2014, DOI:10.3390/ijerph110201573

[15]. Ebrahim Ghafar-Zadeh, Bahareh Gholamzadeh, Giancarlo Ayala-Charca, Parastoo Baghaei, "Toward spirometry-on-chip: design, implementation and experimental results", Microsystem Technologies 23(10), October 2017 DOI:10.1007/s00542-016-3200-0

[16]. Idir Mellal, Mourad Laghrouche, Hung Tien Bui, "Field Programmable Gate Array (FPGA) Respiratory Monitoring System Using a Flow Microsensor and an Accelerometer", Measurement Science Review, 17, 2017, No. 2, 61-67

[17]. Idir Mellal, Youcef Fouzar, Laghrouche Mourad, Jumana Boussey, "FPGA-Based Error Correction in MEMS Sensors: Case Study of Respiration Monitoring System", In book: Recent Advancements in ICT Infrastructure and Applications (pp.65-89), June 2022, DOI:10.1007/978-981-19-2374-6_3

[18]. https://aneskey.com/chapter-13-spirometry/

## APPENDIX

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_signed.ALL;    -- step = step + 1
use ieee.std_logic_unsigned.ALL;

entity DE10_Lite_spirometry is
generic(ClockFrequencyHz : integer:=50000000);

port
(
rst : in std_logic; --SW8
nRst    : in std_logic; -- Negative reset
Seconds : inout integer;
Ticks : inout integer;
led1: out std_logic;--1-5(0-4) LEDs light up when air flow values remain equal or higher than upper limit
led2: out std_logic;
led3: out std_logic;
led4: out std_logic;
led5: out std_logic;
led6: out std_logic;--6-10(5-9) LEDs light up when air flow values are equal or lower than lower limit
led7: out std_logic;
led8: out std_logic;
led9: out std_logic;
led10: out std_logic;
led_blue: buffer std_logic; --air flow low zone values
led_red: buffer std_logic;--air flow high zone values
led_green: buffer std_logic;--air flow medium zone values
led_blue_out: out std_logic;-- air flow low zone values
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026
DOI: 10.17148/IJIREEICE.2026.14206

```
led_red_out: out std_logic;-- air flow high zone values
led_green_out: out std_logic;--air flow medium zone values
led_white: buffer std_logic;--ratio FEV1/FVC< 0.7-lungs problem
led_yellow: buffer std_logic;--ratio FEV1/FVC>= 0.7-no lungs problem
led_white_out: out std_logic;--FEV1/FVC< 0.7-lungs problem
led_yellow_out: out std_logic;--ratio FEV1/FVC>= 0.7-no lungs problem
buzzer:out std_logic; --rings on when FEV1/FVC< 0.7
Vr: buffer integer;
ratio: buffer integer;
FEV1: buffer integer;
FVC: buffer integer;
d2bbuf :buffer integer range 0 to 9;
d1bbuf :buffer integer range 0 to 9;
d0bbuf :buffer integer range 0 to 9;
d2cbuf :buffer integer range 0 to 9;
d1cbuf :buffer integer range 0 to 9;
d0cbuf :buffer integer range 0 to 9;
SW0 : in std_logic;
SW1 : in std_logic;
-- Clocks
ADC_CLK_10: in std_logic;
MAX10_CLK1_50: in std_logic;
MAX10_CLK2_50: in std_logic;
-- KEYs
KEY: in std_logic_vector(1 downto 0);
-- HEX
HEX0: out std_logic_vector(7 downto 0);
HEX1: out std_logic_vector(7 downto 0);
HEX2: out std_logic_vector(7 downto 0);
HEX3: out std_logic_vector(7 downto 0);
HEX4: out std_logic_vector(7 downto 0);
HEX5: out std_logic_vector(7 downto 0);
ARDUINO_IO: inout std_logic_vector(15 downto 0);
ARDUINO_RESET_N: inout std_logic);
-- GPIO
--GPIO: inout std_logic_vector(35 downto 0));
end entity;
architecture DE10_Lite_spirometry_Arch of DE10_Lite_spirometry is
-- Analog to Digital Converter IP core
component myADC is
port
(
clk_clk: in std_logic := 'X';
modular_adc_0_command_valid: in std_logic := 'X';
modular_adc_0_command_channel: in std_logic_vector(4 downto 0) := (others => 'X');
modular_adc_0_command_startofpacket: in std_logic := 'X';
modular_adc_0_command_endofpacket: in std_logic := 'X';
modular_adc_0_command_ready: out std_logic;
modular_adc_0_response_valid: out std_logic;
modular_adc_0_response_channel: out std_logic_vector(4 downto 0);
modular_adc_0_response_data: out std_logic_vector(11 downto 0);
modular_adc_0_response_startofpacket: out std_logic;
modular_adc_0_response_endofpacket: out std_logic;
reset_reset_n: in std_logic
);
end component myADC;
signal modular_adc_0_command_valid: std_logic;
signal modular_adc_0_command_channel: std_logic_vector(4 downto 0);
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026
DOI: 10.17148/IJIREEICE.2026.14206

```
signal modular_adc_0_command_startofpacket: std_logic;
signal modular_adc_0_command_endofpacket: std_logic;
signal modular_adc_0_command_ready: std_logic;
signal modular_adc_0_response_valid: std_logic;
signal modular_adc_0_response_channel: std_logic_vector(4 downto 0);
signal modular_adc_0_response_data: std_logic_vector(11 downto 0);
signal modular_adc_0_response_startofpacket: std_logic;
signal modular_adc_0_response_endofpacket: std_logic;
signal clk_clk: std_logic;
signal reset_reset_n: std_logic;
type state_machines is (sm0,sm1, sm2, sm3, sm4);
signal sm: state_machines;
-- signals to store conversion results
signal ADCIN1,ADCIN4, ADCIN3,ADCIN2: std_logic_vector(11 downto 0);
signal AD1,AD4, AD3,AD2: std_logic_vector(11 downto 0);
-- signal for BCD digits
signal digit2b, digit1b, digit0b: std_logic_vector(3 downto 0);
signal digit2, digit1, digit0: std_logic_vector(3 downto 0);
signal digit2c, digit1c, digit0c: std_logic_vector(3 downto 0);
signal digit5, digit4, digit3: std_logic_vector(3 downto 0);
-- signal to determine how fast the
-- 7-seg displays will be updated
signal cnt: integer;
signal state_LED_right: std_logic;
signal state_LED_left: std_logic;
signal state_Vr: integer;
signal state_ratio: integer;
begin
-- ADC port map
adc1: myADC port map
(
modular_adc_0_command_valid => modular_adc_0_command_valid,
modular_adc_0_command_channel => modular_adc_0_command_channel,
modular_adc_0_command_startofpacket => modular_adc_0_command_startofpacket,
modular_adc_0_command_endofpacket => modular_adc_0_command_endofpacket,
modular_adc_0_command_ready => modular_adc_0_command_ready,
modular_adc_0_response_valid => modular_adc_0_response_valid,
modular_adc_0_response_channel => modular_adc_0_response_channel,
modular_adc_0_response_data => modular_adc_0_response_data,
modular_adc_0_response_startofpacket => modular_adc_0_response_startofpacket,
modular_adc_0_response_endofpacket => modular_adc_0_response_endofpacket,
clk_clk => clk_clk,
reset_reset_n => reset_reset_n
);
clk_clk <= MAX10_CLK1_50;
reset_reset_n <= KEY(0);
-- process for reading new samples
p1: process(reset_reset_n, clk_clk)
begin
if reset_reset_n = '0' then
        sm <= sm0;
elsif rising_edge(clk_clk) then
        case sm is
                when sm0 =>
                        sm <= sm1;
                        modular_adc_0_command_valid <= '1';
                        modular_adc_0_command_channel <= "00001";
                when sm1 =>
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026

DOI: 10.17148/IJIREEICE.2026.14206

```
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00010";
                         ADCIN4 <= modular_adc_0_response_data;
                                sm <= sm2;
                        end if;
                when sm2 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00001";
                                modular_adc_0_command_channel <= "00011";
                                ADCIN1 <= modular_adc_0_response_data;
                                sm <= sm1;
                                sm <= sm3;
                        end if;
                when sm3 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00100";
                                ADCIN2 <= modular_adc_0_response_data;
                                sm <= sm4;
                        end if;
                when sm4 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00001";
                                ADCIN3 <= modular_adc_0_response_data;
                                sm <= sm1;
                        end if;
                when others =>
          end case;
end if;
end process;
process(MAX10_CLK1_50) is
   begin
            if rising_edge(MAX10_CLK1_50) then
         ------ If the negative reset signal is active
         if nRst = '0' then
            Ticks   <= 0;
            Seconds <= 0;
         else
            -- True once every second
            if Ticks = ClockFrequencyHz - 1 then
               Ticks <= 0;
                  Seconds <= Seconds + 1;
            else
               Ticks <= Ticks + 1;
            end if;
         end if;
      end if;
   end process;
-- process for conversion from binary to BCD (analog input voltage)
p3: process(AD2,d2bbuf,d1bbuf,d0bbuf)
variable vin: integer;
variable d2, d1, d0: integer;
begin

vin := to_integer(unsigned(std_logic_vector(to_unsigned(to_integer(unsigned(AD2)) * 500,
32))(31 downto 12)));
d2 := vin / 100;
d1 := vin mod 100 / 10;
d0 := ((vin mod 100) mod 10);
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026
DOI: 10.17148/IJIREEICE.2026.14206

```
digit2b <= std_logic_vector(to_unsigned(d2, 4));
digit1b <= std_logic_vector(to_unsigned(d1, 4));
digit0b <= std_logic_vector(to_unsigned(d0, 4));
d2bbuf<= d2;
d1bbuf<= d1;
d0bbuf<= d0;
end process;
state_Vr<= (d2bbuf*100)+(d1bbuf*10)+(d0bbuf);
Vr<= state_Vr;
-- process for measuring and converting FEV1 and FVC
p4: process(Vr,seconds,d2cbuf,d1cbuf,d0cbuf, FEV1,FVC)
variable d2, d1, d0: integer;
variable t1, t2: integer;
variable Vr1, Vr2: integer;
begin
IF SW1= '1' THEN
t1 := seconds;
END IF;
IF seconds= (t1 +1) THEN
Vr1 := Vr;
FEV1<= Vr1;
END IF;
IF seconds= (t1 +4) THEN
Vr2 := Vr;
FVC<= Vr2;
END IF;
IF SW1= '1' THEN
d2 := (FEV1/FVC) / 100;
d1 := (FEV1/FVC) mod 100 / 10;
d0 := (((FEV1/FVC) mod 100) mod 10);
digit2c <= std_logic_vector(to_unsigned(d2, 4));
digit1c <= std_logic_vector(to_unsigned(d1, 4));
digit0c <= std_logic_vector(to_unsigned(d0, 4));
d2cbuf<= d2;
d1cbuf<= d1;
d0cbuf<= d0;
ELSE
d2 := 0 / 100;
d1 := 0 mod 100 / 10;
d0 := ((0 mod 100) mod 10);
digit2c <= std_logic_vector(to_unsigned(d2, 4));
digit1c <= std_logic_vector(to_unsigned(d1, 4));
digit0c <= std_logic_vector(to_unsigned(d0, 4));
d2cbuf<= d2;
d1cbuf<= d1;
d0cbuf<= d0;
END IF;
end process;
state_ratio<= (d2cbuf*100)+(d1cbuf*10)+(d0cbuf);
ratio<= state_ratio; -- ratio 70 corresponds to 0.7
-- determine how fast the 7-seg displays will be updated
p5: process(reset_reset_n, clk_clk)
begin
if reset_reset_n = '0' then
cnt <= 0;
elsif rising_edge(clk_clk) then
if cnt < 20_000_000 then
cnt <= cnt + 1;
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026
DOI: 10.17148/IJIREEICE.2026.14206

```
else
cnt <= 0;
AD1 <= ADCIN1;
AD2 <= ADCIN2;
AD3 <= ADCIN3;
AD4 <= ADCIN4;
end if;
end if;
end process;
process (Vr,MAX10_CLK1_50, state_LED_right,led_blue,led_red,
led_green, state_LED_left )

begin
--50000000 Ticks= 1sec and Vr=10 corresponds to 0.1V input
IF Vr>=40 THEN
led_red<='1';
state_LED_right<='1';
else
led_red<='0';
state_LED_right<='0';
end if;
IF Vr<25 THEN
led_blue<='1';
state_LED_left<='1';
else
led_blue<='0';
state_LED_left<='0';
end if;
IF Vr<40 AND Vr>=25 THEN
led_green<='1';
else
led_green<='0';
end if;
end process;
led_blue_out<=led_blue;
led_red_out<=led_red;
led_green_out<=led_green;
led1<=state_LED_right;
led2<=state_LED_right;
led3<=state_LED_right;
led4<=state_LED_right;
led5<=state_LED_right;
led6<=state_LED_left;
led7<=state_LED_left;
led8<=state_LED_left;
led9<=state_LED_left;
led10<=state_LED_left;

Process(ratio, led_yellow, led_white)
BEGIN
IF ratio< 70 THEN
led_white<='1';
ELSE
led_white<='0';
END IF;
IF ratio>= 70 THEN
led_yellow<='1';
ELSE
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026
DOI: 10.17148/IJIREEICE.2026.14206

```
led_yellow<='0';
END IF;
END PROCESS;
led_yellow_out<=led_yellow;
led_white_out<=led_white;

-- (FEV1/FVC)< 0.7 buzzer sounds
Process(MAX10_CLK1_50,ratio)
variable i : integer := 0;
BEGIN
IF ratio<70 AND SW1='1' THEN -- buzzer sounds
if MAX10_CLK1_50'event and MAX10_CLK1_50 = '1' then
if i <= 50000000 then
i := i + 1;
buzzer <= '1';
elsif i > 50000000 and i < 100000000 then
i := i + 1;
buzzer <= '0';
elsif i = 100000000 then
i := 0;
end if;
end if;
else
buzzer <= '0';
end if;
end process;
process(SW0,digit2b,digit1b,digit0b,digit2c,digit1c,digit0c)
begin
  IF SW0='0' THEN
  digit2 <= digit2b;--first digit of air flow
  digit1 <= digit1b;--second digit of air flow
  digit0 <= digit0b;--third digit of air flow
  digit5 <= digit2c;--first digit of FEV1/FVC ratio
  digit4 <= digit1c;--second digit of FEV1/FVC ratio
  digit3 <= digit0c;--third digit of FEV1/FVC ratio
  end if;
end process;

WITH digit2 SELECT
HEX2 <= "01000000" WHEN "0000", -- display 0
"01111001" WHEN "0001", -- display 1
"00100100" WHEN "0010", -- display 2
"00110000" WHEN "0011", -- display 3
"00011001" WHEN "0100", -- display 4
"00010010" WHEN "0101", -- display 5
"00000011" WHEN "0110", -- display 6
"01111000" WHEN "0111", -- display 7
"00000000" WHEN "1000", -- display 8
"00011000" WHEN "1001", -- display 9
"11111111" WHEN OTHERS; -- blank display

WITH digit1 SELECT
HEX1 <= "11000000" WHEN "0000", -- display 0
"11111001" WHEN "0001", -- display 1
"10100100" WHEN "0010", -- display 2
"10110000" WHEN "0011", -- display 3
"10011001" WHEN "0100", -- display 4
"10010010" WHEN "0101", -- display 5
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 2, February 2026
DOI: 10.17148/IJIREEICE.2026.14206

```
"10000011" WHEN "0110", -- display 6
"11111000" WHEN "0111", -- display 7
"10000000" WHEN "1000", -- display 8
"10011000" WHEN "1001", -- display 9
"11111111" WHEN OTHERS; -- blank display

WITH digit0 SELECT
HEX0 <= "11000000" WHEN "0000", -- display 0
"11111001" WHEN "0001", -- display 1
"10100100" WHEN "0010", -- display 2
"10110000" WHEN "0011", -- display 3
"10011001" WHEN "0100", -- display 4
"10010010" WHEN "0101", -- display 5
"10000011" WHEN "0110", -- display 6
"11111000" WHEN "0111", -- display 7
"10000000" WHEN "1000", -- display 8
"10011000" WHEN "1001", -- display 9
"11111111" WHEN OTHERS; -- blank display

WITH digit5 SELECT
HEX5 <= "01000000" WHEN "0000", -- display 0
"01111001" WHEN "0001", -- display 1
"00100100" WHEN "0010", -- display 2
"00110000" WHEN "0011", -- display 3
"00011001" WHEN "0100", -- display 4
"00010010" WHEN "0101", -- display 5
"00000011" WHEN "0110", -- display 6
"01111000" WHEN "0111", -- display 7
"00000000" WHEN "1000", -- display 8
"00011000" WHEN "1001", -- display 9
"11111111" WHEN OTHERS; -- blank display

WITH digit4 SELECT
HEX4 <= "11000000" WHEN "0000", -- display 0
"11111001" WHEN "0001", -- display 1
"10100100" WHEN "0010", -- display 2
"10110000" WHEN "0011", -- display 3
"10011001" WHEN "0100", -- display 4
"10010010" WHEN "0101", -- display 5
"10000011" WHEN "0110", -- display 6
"11111000" WHEN "0111", -- display 7
"10000000" WHEN "1000", -- display 8
"10011000" WHEN "1001", -- display 9
"11111111" WHEN OTHERS; -- blank display

WITH digit3 SELECT
HEX3 <= "11000000" WHEN "0000", -- display 0
"11111001" WHEN "0001", -- display 1
"10100100" WHEN "0010", -- display 2
"10110000" WHEN "0011", -- display 3
"10011001" WHEN "0100", -- display 4
"10010010" WHEN "0101", -- display 5
"10000011" WHEN "0110", -- display 6
"11111000" WHEN "0111", -- display 7
"10000000" WHEN "1000", -- display 8
"10011000" WHEN "1001", -- display 9
"11111111" WHEN OTHERS; -- blank display
end architecture;
```