ISSN 2347 - 3983
Volume 8. No. 9, September 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter28892020.pdf
https://doi.org/10.30534/ijeter/2020/28892020

Mixed-signal generator module:Design and Verification in MATLAB and Verilog hardware description language

Phuong H. Lai[1], Duong B. Nguyen[2], Tung V. Nguyen[2], Khoa D. Ta[2], Thien V. Truong[2], and Duc D. Nhu[3]
[1]Dept. of Computing Fundamentals, FPT University, Hanoi, Vietnam;phuonglh17@fe.edu.vn
[2]ICT Department, FPT University, Hanoi, Vietnam;duongnbhe130658@fpt.edu.vn, tungnvhe130151@fpt.edu.vn, khoatdhe130813@fpt.edu.vn, thientvse04522@fpt.edu.vn
[3]Dept. of Multimedia Engineering, Dongguk University, Seoul, South Korea; nhudinhduc@mme.dongguk.edu

ABSTRACT
In this paper, we will discuss the mixed signal generator module in the modern information system. Generally, the mixed signal generator module aims are to store, communicate information in digital form and it can be worked as a transmitter. At first, we will have small review of discrete time signal, its role on a communication system, and the way it can be designed by using MATLAB tool. Then, a proposed algorithm is explained to implement the MATLAB design in Verilog hardware description language. Finally, a survey on a novel application of mixed signal generator module in a current information system is discussed. We hope the paper will help engineers and researchers to have to great reference on current trend, researches, and applications of information system.

Key words: Digital system design, Application specific integrated design-ASIC, MATLAB, Model simulation, Mixed signal, Read only memory-ROM, Look-up table-LUT.

1. INTRODUCTION
Electronic devices can be found everywhere around us, and which have impacted our living every moment [1]. Electronic devices are a complex system which compose of many sub-modules; overall, we can divide them into analogue and digital modules where the digital signal and analogue signals are working there. Signal generator is one of the most important modules and can be found in any analogue and digital modules [2]. In our research, we will focus on digital module and specifically, digital signal generator. There are many purposes of digital signal generator such as driving, testing, troubleshooting, repairing, computing, … though it often has artistic uses as well [3].There are also many different types of signal generator with different applications at varying level of expense and in general, there is no device which is suitable for all possible application [4]. In this research, we propose a procedure of using ROM (READ-ONLY MEMORY) to store a small number of samples; then a controller will be used to generate a full mixed signal.
The proposed research will focus on analysis, design, and implementation of a system on chip design for mixed signal generator module. The study is flexible, open, and useful to be the intellectual logic core which can be reused in future digital system with suitable digital design. The analysis and design are conducted by using MATLAB R2020b. Then, the result of MATLAB design will be used for the implementation and verification by using Model Simulation version 10.4a.
The organization of this study is as follows.In Section 2, we explain the role of discrete signal on information system. The analysis, design, implementation, and verification are discussed in Section 3. The discussion and conclusion are given in Section 4.

2. TIME SIGNAL ON DIGITAL SYSTEM

A. Overview of discrete time signal
In information system, the sensor modules will collect the measurement data from environment such as temperature, humid, pressure, sound, lights, images… and such collected data are called raw data. Since raw data are changing over time, they can be called the time-domain raw data. The period between two continuous measurements is called sampling period, and as corresponding, we also have a parameter sampling frequency. The mathematical expression of raw data can be as following:

$$x_0, x_{\Delta_t}, x_{2\Delta_t}, x_{3\Delta_t}, \dots, x_{N\Delta_t}, \dots$$

where Δ_t is the sampling time, and corresponding sampling frequency is calculated as $\Delta_f = \frac{1}{\Delta_t}$. The raw data are mostly the mixed signals, which are combined of many frequencies signal, or a sine signal with some random noise from environments such as display noise, ...

In MATLAB tool, mixed signals can be simulated by using pre-defined framework. For example, the signal in figure 1 is a sine wave of N = 50 samples, the sampling frequency is 48KHz, signal frequency is 2KHz, and the parameter 'x' denotes the sine signal.

```
clear; clear all;
N    = 50;
n    = 0:1:N-1;
fs   = 48000;     %% Sampling frequency
f1   = 2000;      %% frequency 1
x    = 1.5*sin(2*pi*(f1/fs)*n);
subplot(3,1,1);
t    = (0:1:N-1)*1/fs;
stem(t,x);
```

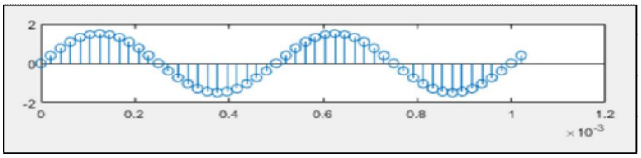

Figure 1: Raw signal in time domain.

B. Sine/Pulse signal generator module on digital system

In communication systems, the signal generators can be used for various purposes such as testing, transmitter, and they can be implemented in various kinds such as we can use analogue circuits, … In our proposed system, we consider the digital processing technique where the preloaded data which are stored in ROM memory and we define the controller to generate expected signal. The advantage of our proposed is that it is flexible and can be reused for numerous kinds of signal rather than we need to change many parameters by using analogue circuits.

Figure 2 shows the block diagram of a full information system where the position of sine/pulse signal generator is in highlight. Signal generator use the preloaded data which can be stored at ROM (read-only memory), EEPROM (electrical erasable programmable read-only memory) and via an adaptive controller which will generate expected discrete signal with pre-defined sampling frequency. The output of that procedure will come to a suitable DAC(digital-analogue converter) module and then, go through an amplifier to be an expected amplitude before working as a transmitter or driving signal.

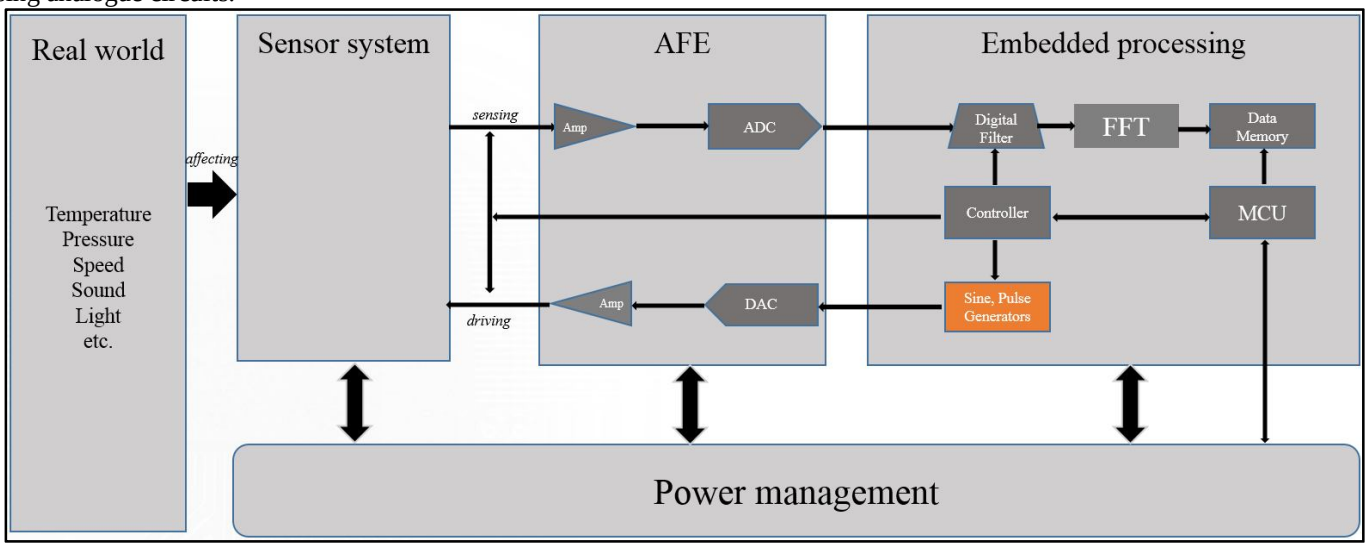

Figure 2: Position of a digital filter module on anoverall digitalsystem.

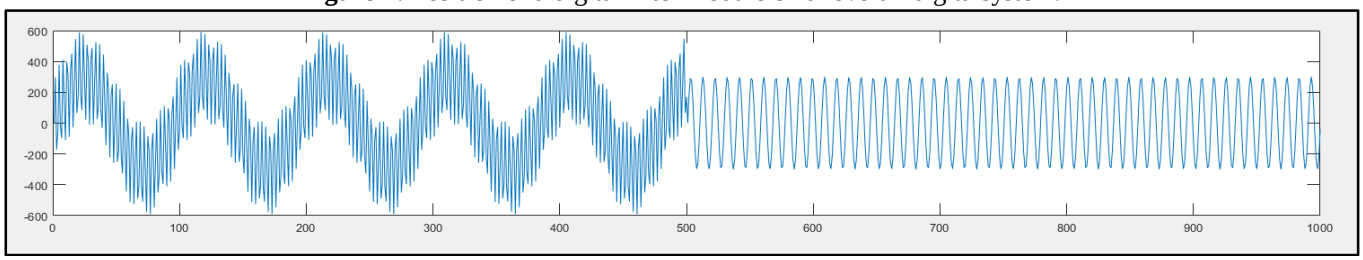

Figure 3: A mixed signal waveform of a design in MATLAB with three frequencies.

3. DESIGN AND VERIFICATION OF SIGNAL GENRATOR MODULE

A. Design of signal and mixed signal by MATLAB tool

MATLAB is a strong tool which provide a mathematical environment for numerous purpose such as mathematical computing, programming, signal visualization, matrices vectors computing, ... In our work, we use MATLAB as a design environment where the mixed signals, their samples with pre-defined sampling frequency are designed.

In figure 4, we use two popular methods in communication technology (frequency division multiplexing (FDM) and time division multiplexing (TDM)) to design a mixed signal. The first and second frequencies will be combined to be a mixed signal with FDM technique (line 11 to line 14), the result is a mixed signal which will be concatenate with third frequency by TDM technique (line 16) to get the expected signal. The waveform of our proposed signal can be seen in figure 3.

```
1  clc; clear all;
2  %Script to generate signal for testbench
3  N = 500;
4  n = 0:N-1;
5
6  fs = 48000;
7  f1 = 500;
8  f2 = 15000;
9  f3 = 5000;
10
11 x1 = round(300*sin(2*pi*(f1/fs)*n));
12 x2 = round(300*sin(2*pi*(f2/fs)*n));
13 y1 = x1 + x2;
14 y2 = round(300*sin(2*pi*(f3/fs)*n));
15
16 y = [y1,y2];
17
18 for i = 1:1:2*N
19     formatSpec = 'x = %16.f; #20833; // Sample(%d)\n';
20     fprintf(formatSpec,y(i),i);
21 end
```

Figure 4: Design for a three frequencies mixed signal.

B. Implementation of signal generator module by VERILOG HDL

Verilog HDL is a hardware description language which is used to model electronic systems and commonly used in implementation and verification of analogue and mixed signal circuits. In our proposed system, we use Verilog to model and verify our MATLAB design. The proposed algorithm is explained in figure 5 where we used MATLAB to design one cycle of mixed signal for each frequency which will output a real data. The real data will be converted into fixed-point data as suitable number representative of Verilog system. The fixed-point data will be store in ROM memory. The purpose of just storing one cycle in ROM is to save the memory space and the sine signal repeat its value after each cycle.

Figure 6 explains a detail step of MATLAB design and Verilog implementation of a sine wave signal which contains just one frequency (already mentioned in figure 1). As can be seen from figure 6, we propose a MATLAB programming to generate a script to define a ROM memory. ROM memory valued is then defined in Verilog file and a controller to repeat it after each cycle. The waveform result is shown in figure 7, it has a similar form as result of visualization result in MATLAB.

In figure 8, we will consider a mixed signal of two frequencies. The MATLAB tool will also be used to calculate two ROM values for each frequency. In addition, two enable clock signals are used to determine the time slot when each frequency is enabling. When first clock is enabling (in high level), signal of first frequency will be represented. The same situation is for signal of second frequency. The waveform of that proposed signal is shown in figure 9. This proposal can be used as a model of a transmitter with a button to determine a signal type in real living application.

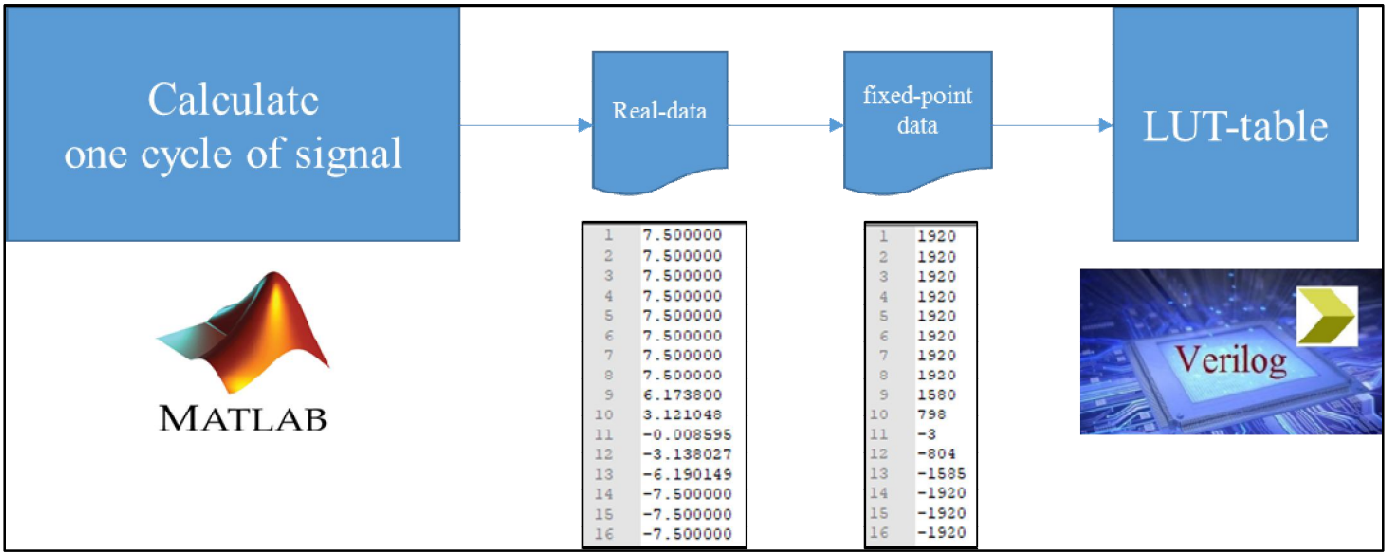

Figure 5: Our proposed algorithm for design and implementation.

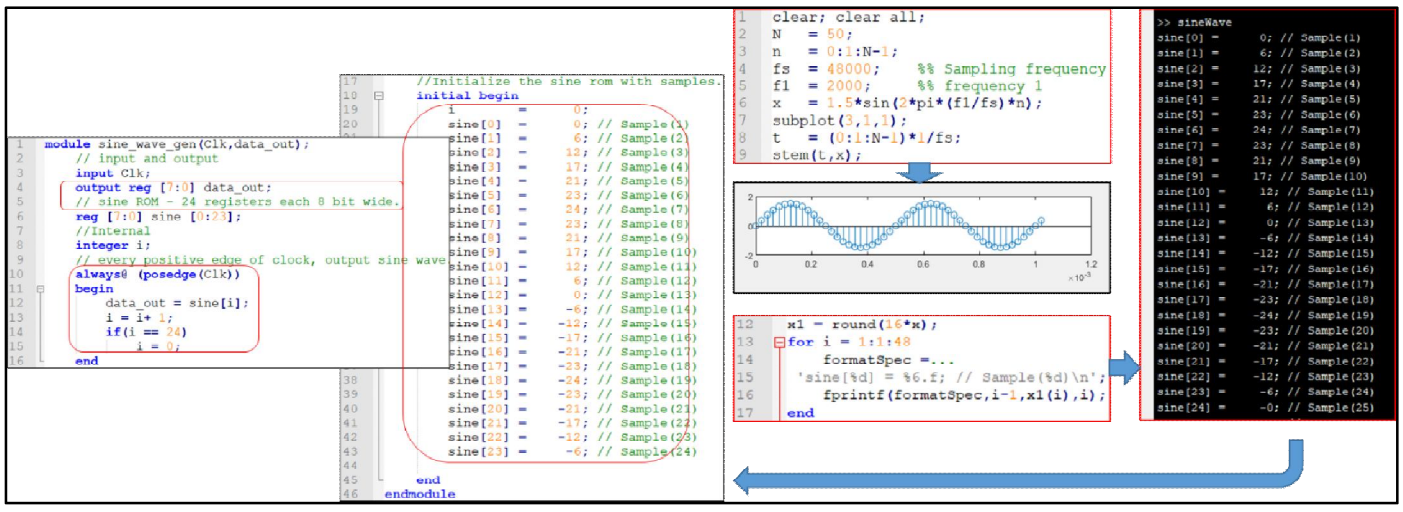

Figure 6: One implementation of a single frequency signal in VERILOG with MATLAB design.

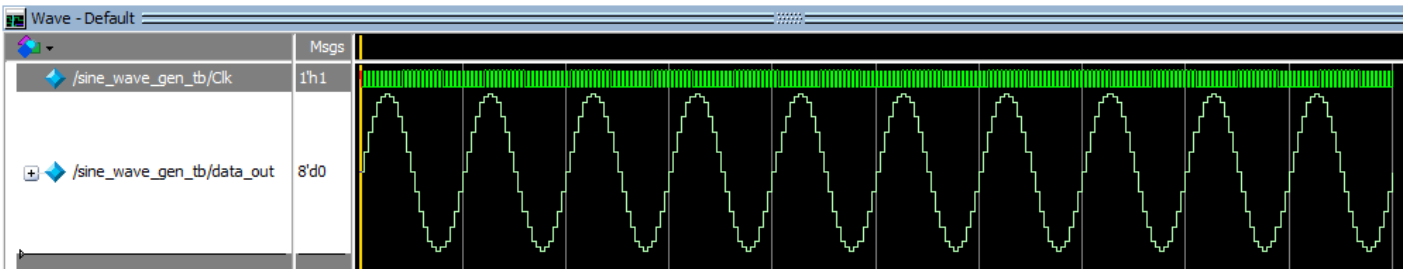

Figure 7: Waveform result of the implementation in figure 6.

```
module MPPTailHover (clk,enableBeacon,enableDigitalLF,MPPsignal_out);
//declare input and output
    input clk;
    input enableBeacon;
    input enableDigitalLF;
    output signed [23:0] MPPsignal_out;
//declare the Beacon ROM - 40 registers each 8 bit wide.
    reg signed [23:0] BeaconData [0:39];
    reg signed [23:0] DigitalLF [0:22];
//Internal signals
    integer i;
    integer i_digLF;
    reg signed [23:0] MPPsignal_out;
```

```
    initial begin
        i_digLF = 0;
    DigitalLF[0]  = 1920;
    DigitalLF[1]  = 1920;
    DigitalLF[2]  = 1920;
    DigitalLF[3]  = 1920;
    DigitalLF[4]  = 1100;
    DigitalLF[5]  = -300;
    DigitalLF[6]  = -1600;
    DigitalLF[7]  = -1920;
    DigitalLF[8]  = -1920;
    DigitalLF[9]  = -1920;
    DigitalLF[10] = -1920;
    DigitalLF[11] = -1920;
    DigitalLF[12] = -1920;
    DigitalLF[13] = -1920;
    DigitalLF[14] = -1920;
    DigitalLF[15] = -1600;
    DigitalLF[16] = -300;
    DigitalLF[17] = 1100;
    DigitalLF[18] = 1920;
    DigitalLF[19] = 1920;
    DigitalLF[20] = 1920;
    DigitalLF[21] = 1920;
    DigitalLF[22] = 1920;
    end
```

```
//At every positive edge of the clock, output a Beaco
    always@ (posedge(clk))
    begin
    if(enableBeacon)
    begin
        MPPsignal_out = BeaconData[i];
        i = i+ 1;
        if(i == 39)
            i = 0;
    end
    else
        if (enableDigitalLF)
        begin
            MPPsignal_out = DigitalLF[i_digLF];
            i_digLF = i_digLF+1;
            if (i_digLF == 22)
                i_digLF = 0;
        end
        else
            MPPsignal_out = 0;
    end
endmodule
```

```
//Initialize the BeaconData rom with samples.
    initial begin
        i = 0;
    BeaconData[0] =  1920;BeaconData[1] =  1920;BeaconData[2] =  1920;BeaconData[3] =  1920;
    BeaconData[4] =  1920;BeaconData[5] =  1920;BeaconData[6] =  1920;BeaconData[7] =  1920;
    BeaconData[8] =  1580;BeaconData[9] =  798;BeaconData[10] =  -3;BeaconData[11] = -804;
    BeaconData[12] =-1585;BeaconData[13] =-1920;BeaconData[14] =-1920;BeaconData[15] =-1920;
    BeaconData[16] =-1920;BeaconData[17] =-1920;BeaconData[18] =-1920;BeaconData[19] =-1920;
    BeaconData[20] =-1920;BeaconData[21] =-1920;BeaconData[22] =-1920;BeaconData[23] =-1920;
    BeaconData[24] =-1920;BeaconData[25] =-1920;BeaconData[26] =-1920;BeaconData[27] =-1920;
    BeaconData[28] =-1577;BeaconData[29] = -798;BeaconData[30] =  3;BeaconData[31] =  804;
    BeaconData[32] = 1585;BeaconData[33] = 1920;BeaconData[34] = 1920;BeaconData[35] = 1920;
    BeaconData[36] = 1920;BeaconData[37] = 1920;BeaconData[38] = 1920;BeaconData[39] = 1920;
    end
```

Figure 8: One implementation of a mixed signal with two frequencies in VERILOG.

Figure 9: Waveform result of the implementation in figure 8.

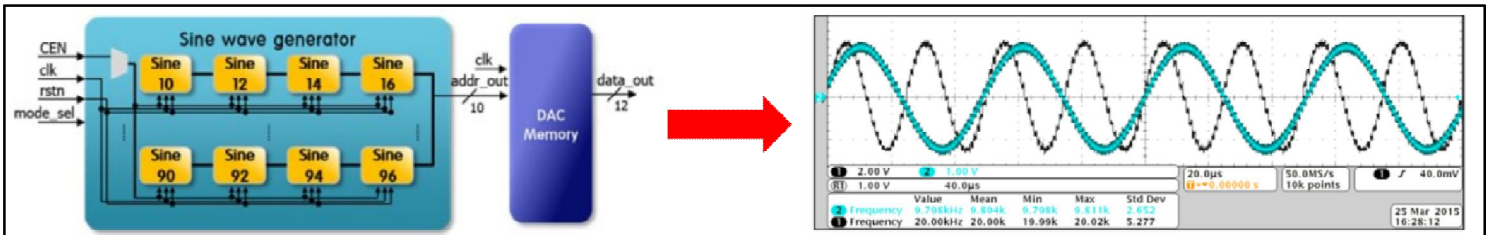

Figure 10: A multiple-frequencies sine wave generator module.

C. Discussion on mixed signal generator module

The main methodology of this paper is to use MATLAB tool to design ROM memories. Then, the data of ROM will be used in Verilog HDL for implementation and verification with an adaptive controller to generate expected signal for transmitter or driving signals.

This procedure can be used in numerous applications. In CORDIC algorithm, itsROM data was pre-defined by MATLAB tool, and the algorithm is implemented by Verilog HDL. We will discuss and analysis CORDIC algorithm in a future paper. At current time, as a development of human-machine interactions, active pen, touch screen, and their communication are hot trends of researches. There are many proposal protocols have been discussed and for a large touch screen, concurrent driving technique is the most effective solution [5]. Inside that protocol, a multiple-frequency sine way generator module is one of the most important modules. As can be seen in figure 10, we just need to store finite samples, then an adaptive controller will choose a suitable sample with different clock timing to generate all type of sine wave generator.These sines wave generators will be used as driving signals for touch screen. We will discuss that problem soon in our future research paper.

4. CONCLUSION

The paper has proposed the design and implementation of mixed signal generator module. The MATLAB tool is used to design the parameters and these parameters are used for implementation and verification which are conducted by Verilog HDL. Some examples with detail analysis and source codes are given. I hope this short paper will be a good reference for researchers, students, and engineers in their design and implementation works. All the comments are welcome and MATLAB, Verilog, test bench source codes will be shared as reasonable request.

ACKNOWLEDGEMENT

This work is supported by FPT University, Hanoi, Vietnam. We would like to thank Dr. Nguyen Manh Duc (G2Touch Company, Republic of Korea) for his useful suggestions.

Conflict of Interest

There is no conflict of interest to publish this research. All authors contribute equally on this research.

REFERENCES

1. Nguyen, D.M., Kim, S. "**A novel construction for quantum stabilizer codes based on binary formalism**". Int. J. of Modern Phys. B. vol. 33(8), 2020.
2. S. Santhosh. et. al. "**Bio Impedance Signal Analysis with Variation in Input Signal Frequency** ".Int. J. of Advanced Trends in Computer Science and Engineering, vol. 9(4), 2020.
3. Hieu V.D.et. al. "**Design and verification of novel classical error control codes using VERILOG Hardware Description Language**". Int. J. of Advanced Trends in Computer Science and Engineering, vol. 9(4), 2020.
4. A. Don M.Africa.et. al. "**Development of a Real-time ECG signal transmission monitoring Algorithm**". Int. J. of Emerging Trends in Engineering Research, vol. 8(8), 2020.
5. J. Kim. et. al. "**Design of a Frequency Division Concurrent Sine Wave Generator for an Efficient Touch screen Controller SoC**". IEEE Int. Symp. on Consumer Electronics (ISCE), 2015.