ISSN (Print) : 2319-8613
ISSN (Online) : 0975-4024

Realisation of Digital Circuits Systems Using Embedded Function on MATLAB

Mittarpal [#1], Naresh Kumar [*2], Sunita Rani [#3]

[#] Department of Electronics and Communication, Bhagat Phool Singh Mahila Vishwavidyalya
Khanpur Kalan, 131305, Sonipat, Haryana, India

[1] mpjangra80@gmail.com

[3] er.sunitarani@gmail.com

[*] Electronics and Communication Engineering, SGT University, Chandu Budera-122505,
Gurugram, Haryana, India

[2] engg.tamwar86@gmail.com

***Abstract*—Digital circuit systems are the backbone of the digital world and play a crucial role in diverse field technology applications viz. computer, mobile, home appliances, in the military, air forces, navy, medical and space and much more. In the present paper, the realization of digital circuit systems using Matlab and Simulink is presented. The aim of this paper is to guide the searchers about the methodology for design digital systems using Matlab and Simulink tools. In era of digital world, the understanding of realization of digital electronic circuits using simulation approach will play guiding role in the research**

Keyword-Logic Gates, Digital circuit systems, converters, Matlab, Simulink, Matlab Function

I. INTRODUCTION

This paper addresses various digital devices and developing their MATLAB function script implementation using embedded matlab function in Simulink model environment. Matlab stands for matrix laboratory, is a general purpose, scientific computing integrated environment, commercial software package. It has editor window for Matlab script and Simulink (graphical environment) to solve a simple mathematical problem to highly complex research-oriented problems. Matlab Simulink, its toolboxes, and block sets, are well known and a leading mathematical application package in various scientific disciplines. It is a high-level computation and simulation language that allows easy and reliable manipulation of vectors and matrices. The latest version of Matlab with data structures, excellent graphics, Graphical user interface (GUI), Simscape and Simulink, is an invaluable tool for research level and implementation.

Matlab can apply to solve different problems in diverse engineering fields [9-23]. Engineering simulation, using Simulink graphical programming plays a crucial role in understanding and assessing the operation of a physical system. Simulink built add-on Matlab, is a very interactive tool for modelling, simulating and analyzing multi-domain systems. Matlab is an ideal tool for qualitative and quantitative analysis of generic and particular problems related to science and technology. It has found the huge scope in laboratory projects. Hence, it has been adopted for teaching a variety of courses in science and technology. The benefits of Matlab and Simulink have been well documented by several researchers [1-8].

This paper proposes the methodology in Matlab, to realize digital circuit system (DCS) on text-based description function, m.file and graphical programming Simulink model using embedded Matlab function. In current work, the graphical Simulink model approach uses block diagram realization with the MATLAB function script. Function script implementation is a text based m.file description of the digital system under reference, which can be written in any text editor for the output of DCSs.

II. DIGITAL CIRCUIT SYSTEMS

Digital circuit systems are the backbone of current technology and innovations, and these play a pivotal role in data acquisition and analysis, control, home and industry appliances, military, space, education, health, automation, automobiles, mobile, internet, radio, television, mobile and much more, mention few.

A. Binary Number System

Digital systems operate on the binary information and it's based on the binary number system. It consists two number as 0 (low potential) and 1 (high potential), known as bits.

B. Fundamental Circuits

The logic gates are the fundamental building blocks for digital circuit systems. These are OR, AND, NOT, XOR, NOR, NAND, XNOR; and their characteristics table in form of truth table with two input A, B and one output as Y is depicted in Table I.

TABLE I
Truth table for all logic gates

Input		Output						
		OR	AND	NOT (m)	XOR	NOR	NAND	XNOR
A	B	Y	Y	Y	Y	Y	Y	Y
0	0	0	0	1	0	1	1	1
0	1	1	0	1	1	0	1	0
1	0	1	0	0	1	0	1	0
1	1	1	1	0	0	0	0	1

C. BCD to Seven Segments Decoder

To visualize the output data in digital format at the output stage, it is required to represent output data into the numeric form. For which BCD to7-segment driver is required that converts the binary coded decimal data into 0-9 numeric form. Seven segment elements are depicted in Fig.1.

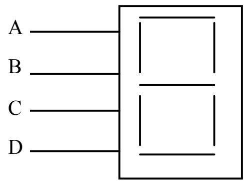

Fig.1. Seven Segments with BCD Input

The all the possible combinations for the BCD to 7- segment display driver, are described in Table II. That will be further utilized to determine the expressions for outputs.

TABLE II
Truth table of BCD to 7-Segments Converter

BCD Input				7-Segment code						
A	B	C	D	A	b	C	D	E	F	D
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	0	1	1
0	1	0	0	1	1	0	1	1	0	0
0	1	0	1	1	1	0	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	0	1	1	0	0	0	0
1	0	0	1	0	1	1	1	0	1	1

D. BCD to Excess-3 Code

In digital systems, another important code is excess-3 code. So it's very useful to design the driver for BCD to Excess 3. The Eexcess-3 code is a self-complementary code. The truth table of BCD to Excess-3 code-convertor is shown in Table III.

TABLE III
Truth Table of BCD to Excess-3 Code

BCD Input Code				Excess-3 Output Code			
A	B	C	D	w	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	1
1	1	1	1	1	0	1	0

E. Parity Bit Checker

In order to one-bit error detection, one-bit is added with the information during transmission, known as parity or check bit. On the basis of a number of one's in binary data are either odd or even number, a respective parity bit is either odd parity or even parity. Truth table odd parity bit checker is depicted in Table IV.

TABLE IV
Truth Table Odd Parity bit Checker

4-bit Information				Additional Parity bit
A	B	C	D	P
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

III. METHODS OF REALIZATION OF LOGIC GATES IN SIMULATION ENVIRONMENT

A. M-File In Matlab

There are two types of M-file in Matlab, depicted in Fig.2. M-files can be a script file that executes a sequence of command or a function file that accept input argument and returns output argument.

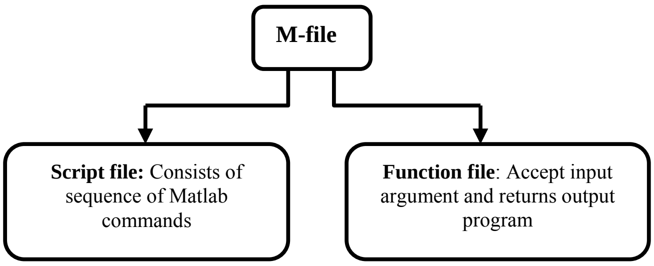

Fig.2. Type of Matlab Codes

B. Steps for Constructing a Script File

The steps for constructing a program in Matlab are as follows:
Step 1. Open the Matlab desktop via the window start menu or by double clicking on the Matlab icon on the desktop.
Step 2. Click on file new script this open new M-file Script window.
Step 3. Type the M-file script into the script editor window.
Step 4. Save the script by clicking on the save icon in the icon toolbar or clicking on file in the menu bar and selecting save in the drop down menu. In the dialogue box that appears, select the folder where the script is to reside and type in the file name of your own choosing. It is the best to use a folder that contains only own MATLAB script
Step 5. Before running the script, you need to go to the current folder box at the top of the MATLAB desktop, clicking on down arrow and in the drop down menu, selecting (or browsing to) the folder that contains your new script
Step 6. Compile and run the MATLAB script from the script window either by clicking on the save and run green arrow in the icon toolbar or alternatively from the command window by typing the script name (without the .m extension) in the MATLAB command prompt.
Step 7. Observe the result of execution/output in command window/figure window

C. Function File

The general structure of function file is shown below:

```
Function [out_1, out_2........] = function_name [in_var1, in_var2.......]
        %H1 line
        %Help text
        Function body
End
```

The first line containing keyword function followed by output parameters enclosed in square bracket followed by function name and input arguments enclosed in parenthesis. In order to execute a function file, firstly go to the Matlab command prompt and type the name of function. The input variables are supplied to the function.

D. Matlab FCN Simulink

The steps of embedded Simulink function are given as below:

Step 1. OpenSsimulink environment either clicking on Simulink icon on the toolbar of Matlab window.
Step 2. Drag Matlab FCN from user defined function library of Simulink.
Step 3. Enter sources and sink elements
Step 4. Connect sources, sink and Matlab function blocks
Step 5. Double click on the Matlab function, open new a editor window.
Step 6. Enter Matlab function script of given digital problem.
Step 7. Save code and run the script.
The Simulink model requires basic understanding of setting of configuration parameters by clicking none on automatic solver parameters selection none. It is also the necessity to set solver and defining digital input as logical.
In the forthcoming section, basics of digital fundamentals and circuits realized using Matlab codes, function scripts, embedded Matlab function of Simulink.

IV. SIMULATION IMPLEMENTATION OF DCSS

A. *Implementation Of BCD to Seven Segment Convertor*

MATLAB Script for BCD to Seven segment convertor is given in Fig.3.

```
% MATLAB script: DCS1bcdsevensegment.m
% MATLAB script to simulate BCD into seven segments
clc;                                          %clear the command window
clear   all;                                  %clear all the variables
close   all;                                  %close all figure windows
clc;                                          % to clear the command window
A=logical (0   0   0   0  0  0  0  0  1  1);  % to define logical input
B=logical (0   0   0   0  1  1  1  1  0  0);  % to define logical input
C=logical (0   0   1   1  0  0  1  1  0  0);  % to define logical input
D=logical (0   1   0   1  0  1  0  1  0  1);
for m=1:9                                     % loop initialization
   a=~A&~B|(A&C)|B|C;
   b=~C|A&B|~A&~B;
   c=A|~B|C;
   d=~A&~B|~B&C|D|~A&B|~A&B&C;
   e=~A&~C|~A&B;
   f=D|~A&C|~A&~B|~B&C;
   g=~A&C|~B&C|D|~D&C;
end
disp ('Truth Table of BCD to Seven Segments S')                % display the text
Disp ('A     B      C      D     a     b     c     d     e     f     g')      % display the text
Table= [A; B; C; D; a; b; c; d; e; f; g];                        % table contents
Fprintf ('%i   %i   %i    %i    %i    %i    %i    %i    %i    %i     %i\n', table)    %to write the data in
```

Fig.3. MATLAB Script for BCD to Seven Segment Convertor

The Execution Matlab program script DCS1bcdsevensegment.m result is given in Fig.4.

Truth Table of BCD to Seven Segments S

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	0	1	1
0	1	0	0	1	1	0	1	1	0	0
0	1	0	1	1	1	0	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	0	1	1	1	1	1
1	0	0	0	0	1	1	0	0	0	0
1	0	0	1	0	1	1	1	0	1	1

Fig.4. Simulation Output of DCS1 BCD seven segment. m

A. BCD INTO EXCESS -3 CODES

The Matlab script is used to convert the BCD into Excess -3 codes, shown in Fig.5.

```
% MATLAB script: DCS1bcdexcess3.m
% MATLAB script to simulate half adder
clc;                                                   %clear the command window
clear   all;                                           %clear all the variables
close   all;                                           %close all figure windows
A=logical (0  0  0  0 0 0 0 0 1 1 1 1 1 1 1 1);    % to define logical input
B=logical (0  0  0  0 1 1 1 1 0 0 0 0 1 1 1 1);    % to define logical input
C=logical (0  0  1  1 0 0 1 1 0 0 1 1 0 0 1 1);    % to define logical input
D=logical (0  1  0  1 0 1 0 1 0 1 0 1 0 1 0 1);
for m=1:16                                             % loop initialization
   z=~D;
   y= (C&D)|(~C&~D) ;
   x=~B&(C|D)|B&~(C|D);
   w=A|B&(C|D);
end
Disp ('Truth Table of BCD to Excess -3 code convertor')          % display the text
Disp ('A    B    C    D    w    x    y    z')                        % display the text
Table= [A; B; C; D; w; x; y; z];                                 % table contents
fprintf('%i   %i   %i   %i   %i   %i   %i   %i\n', table)         %to write the data in text
```

Fig.5. Matlab Code of BCD into Excess -3 Codes

The Execution of Matlab program script DCS1bcdexcess3.m result is given in Fig.6.

Truth Table of BCD to Excess -3 code convertor

BCD				Excess-3				
A	B	C	D	v	w	x	y	z
0	0	0	0	0	0	0	1	1
0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	1	0	1
0	0	1	1	0	0	1	1	0
0	1	0	0	0	0	1	1	1
0	1	0	1	0	1	0	0	0
0	1	1	0	0	1	0	0	1
0	1	1	1	0	1	0	1	0
1	0	0	0	0	1	0	1	1
1	0	0	1	0	1	1	0	0
1	0	1	0	0	1	1	0	1
1	0	1	1	0	1	1	1	0
1	1	0	0	0	1	1	1	1
1	1	0	1	1	0	0	0	0
1	1	1	0	1	0	0	0	1
1	1	1	1	1	0	0	1	0

Fig.6. Simulation Output of DCS1bcdexcess3.m

The MATLAB Script for parity bit generation is given Fig.7 and shown below:

```
% MATLAB script: DCS1paritybitchecker.m
% MATLAB script to simulate half adder
clc;                                                  %clear the command window
clear   all;                                          %clear all the variables
close   all;                                          %close all figure windows
clc;                                                  % to clear the command window
A = logical (0  0  0  0 0 0 0 0 1 1 1 1 1 1 1);       % to define logical input
B = logical (0  0  0  0 1 1 1 1 0 0 0 0 1 1 1 1);     % to define logical input
C = logical (0  0  1  1 0 0 1 1 0 0 1 1 0 0 1 1);     % to define logical input
D = logical (0  1  0  1 0 1 0 1 0 1 0 1 0 1 0 1);
for m=1:16                                            % loop initialization
   p=xor(xor(xor(A,B),C),D);
end
Disp ('Truth Table Odd Parity bit Checker')           % display the text
Disp ('A     B       C       D      p')                % display the text

Table= [A;B;C;D;p];                                   % table contents

fprintf('%i     %i       %i       %i      %i \n', table)   %to write the data in text
```

Fig.7. MATLAB Script for parity bit checker

The Execution of Matlab program script DCS1paritybitchecker.m result is depicted in Fig.8.

Truth Table Odd Parity bit Checker

A	B	C	D	Parity (P)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Fig.8. Simulation Output of DCS1paritybitchecker.m

To realize the implementation of basic logic gates using Matlab function, necessity of develop matlab code. The logic gate OR, AND, XOR are implemented using matlab function in Simulink is shown in Fig.9. The Matlab script to implement the logic gates is depicted in Fig.10

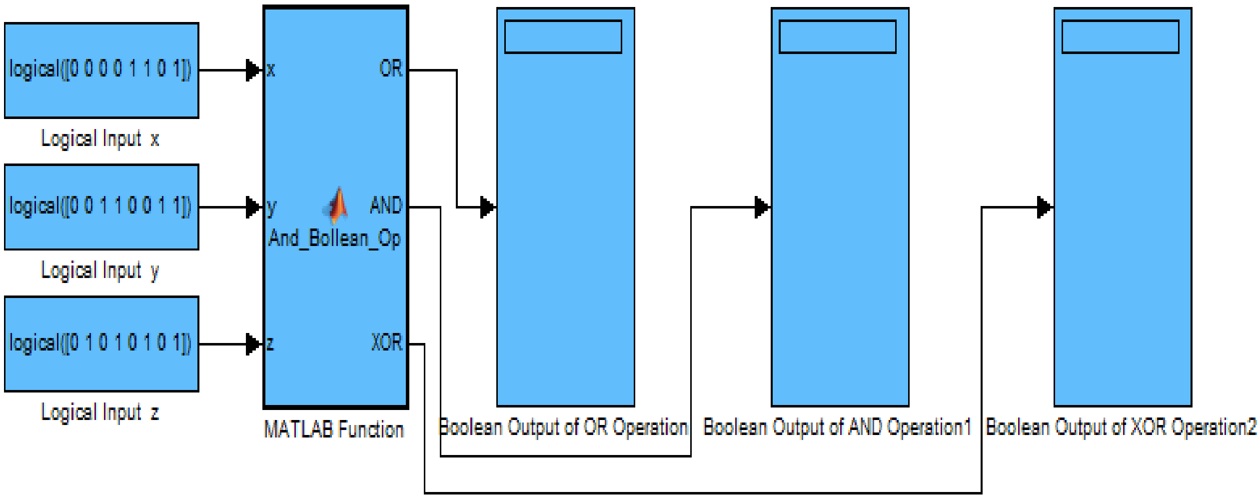

Fig.9. Realization of Basic Logic Gates

```
Function [OR, AND, XOR] =And_Bollean_Op(x, y, z)
% Usage:
% Compute and Boolean operation between x, y,
z %***********************************************************
% Define input variable of function And_Bollean_Opr1
%***********************************************************
% Inputs:
% Logical input x
% Logical input y
% Logical input z
%***********************************************************
% Output variable of function And_Bollean_Opr1
%***********************************************************
%Output
%Logical Function f(x, y, z)
 %***********************************************************
% Compute output of function And_Bollean_Opr1
%***********************************************************
for m=1:8                          % loop initialization
   AND=x&y&z;
   OR=x|y|z                        % control statement
   XOR=xor (z, (xor(x, y)))
End                              % to terminate the loop
%***********************************************************
```

Fig.10. Matlab Script for AND, OR, XOR logical Gates

In Fig.11, BCD to seven segment Simulink model using embedded Matlab function is depicted.

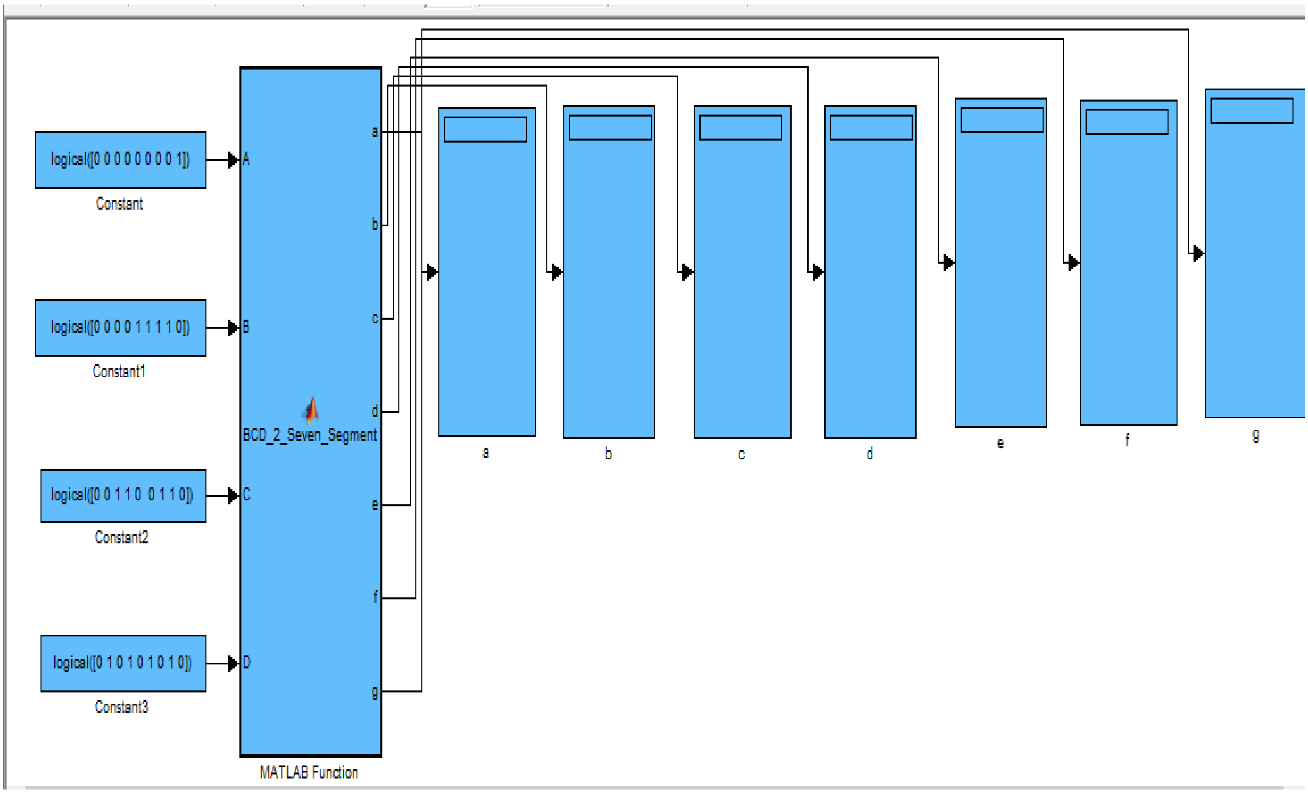

Fig.11. Realization of BCD to Seven Segment Converter

The Matlab code in form of function to convert BCD into seven segment code is shown in Fig.12. This code is used in embedded matlab function.

```
function [a,b,c,d,e,f,g]=BCD_2_Seven_Segment (A,B,C,D)
% Usage:
%Convert BCD to seven segment code ***********************************
% Define input variable of function BCD2sevensegment******************
% Inputs:
% A: Logical i/p % B: Logical i/p % C: Logical i/p % D: Logical i/p ***
% Output variable of function BCD2sevensegment*************************
% Output:
% Element of seven segments
% a, b, c, d, e, f, g
for m=1:9                    % loop initialization
   a=~A&~B|(A&C)|B|C;
   b=~C|A&B|~A&~B;
   c=A|~B|C;
   d=~A&~B|~B&C|D|~A&B|~A&B&C;
   e=~A&~C|~A&B;
   f=D|~A&C|~A&~B|~B&C;
   g=~A&C|~B&C|D|~D&C;
end
```

Fig.12. MATLAB Script for BCD to Seven Segment Convertor

For the realization of half adder, subtractor and full adder, subtractor, the Simulink model using embedded Simulink function is designed and it Matlab function code is shown in Fig.13 and model is represented in Fig.14.

```
Function [a, b, c, d, e, f, g, h] =Combinational_Digital_Ctk(x1, y1, x, y, z)
% Usage:
%Combinational_digital_ctk
%******Define input variable of function Combinational_Digital_Ctk*******
% Inputs:
% x1: Define logical input x1 % y1: Define logical input y1
% Usage: Full adder
% x: Define logical input x
% y: Define logical input y
% z: Define logical input z
%****** Output variable of function Combinational_Digital_Ctk**************
%Output:
% Half-adder % Sum 'a' %Carry 'b'
% Full-adder   % Sum 'c' %Carry'd'
% Half-subtractor  %Difference, e   % Borrow, f
% Full-subtractor   %Difference, g   % Borrow, h
for m=1:4                              % loop initialization
   a=xor(x1, y1);                      % control statement
   b=x1&y1;                            % control statement
End
For m=1:8                              % loop initialization
  c=xor ((xor(x, y)), z);              % control statement
   d=x&y| (z& (xor(x, y)));            % control statement
End
For m=1:4                              % loop initialization
   e= (~x1) &y1|x1& (~y1);             % control statement
   f=~x1&y1;                           % control statement
End
For m=1:8                          % loop initialization
   g=~x&~y&z|~x&y&~z|x&~y&~z|x&y&z;       % g is difference
   h=~x&y|~x&z|y&z;                    % h is borrow
End
```

Fig.13. MATLAB Script for Combinational circuits - Adder and Subtractor

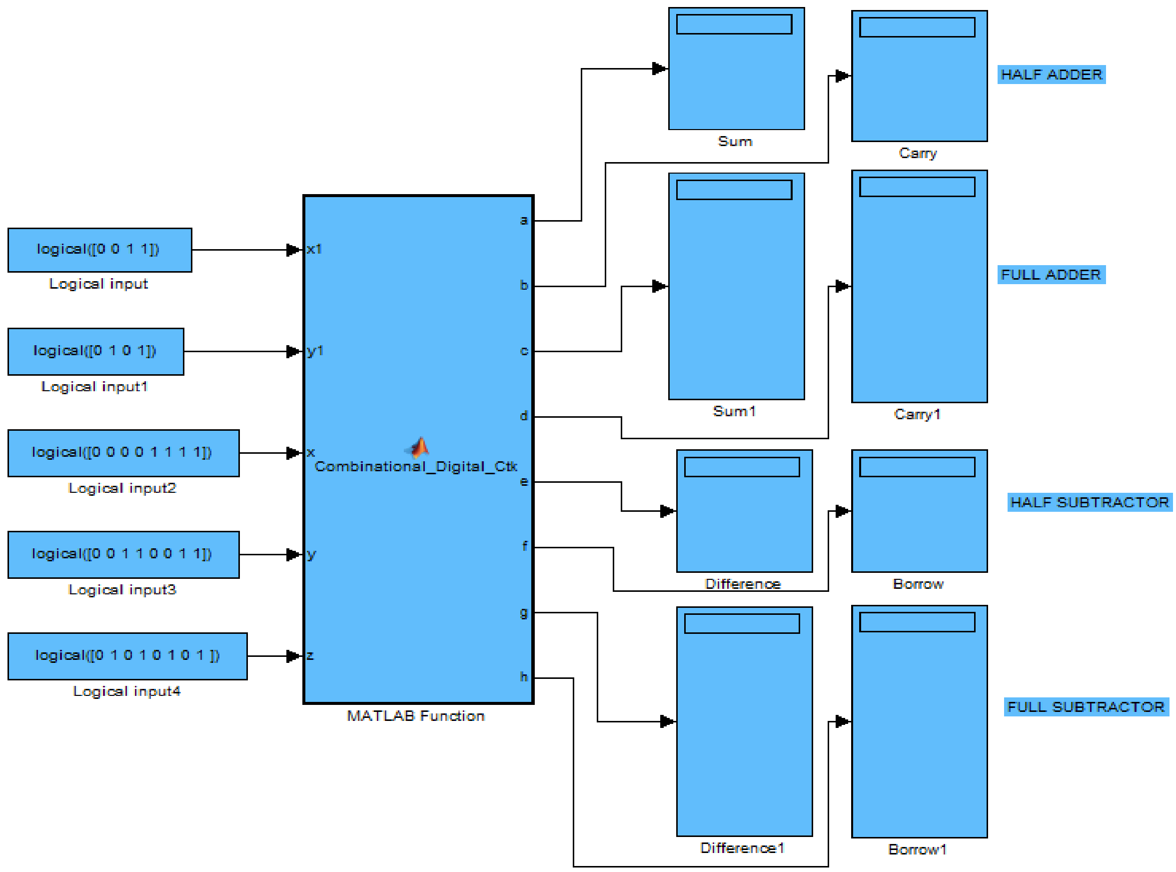

Fig.14. Realization of Half and Full Adder and Half and Full Subtractor

The conversion of BCD code into Excess-3 code using Matlab, Simulink model is depicted in Fig.15. This model is developed using embedded matlab function. The code of conversion from BCD to excess-3 is shown in Fig.16.

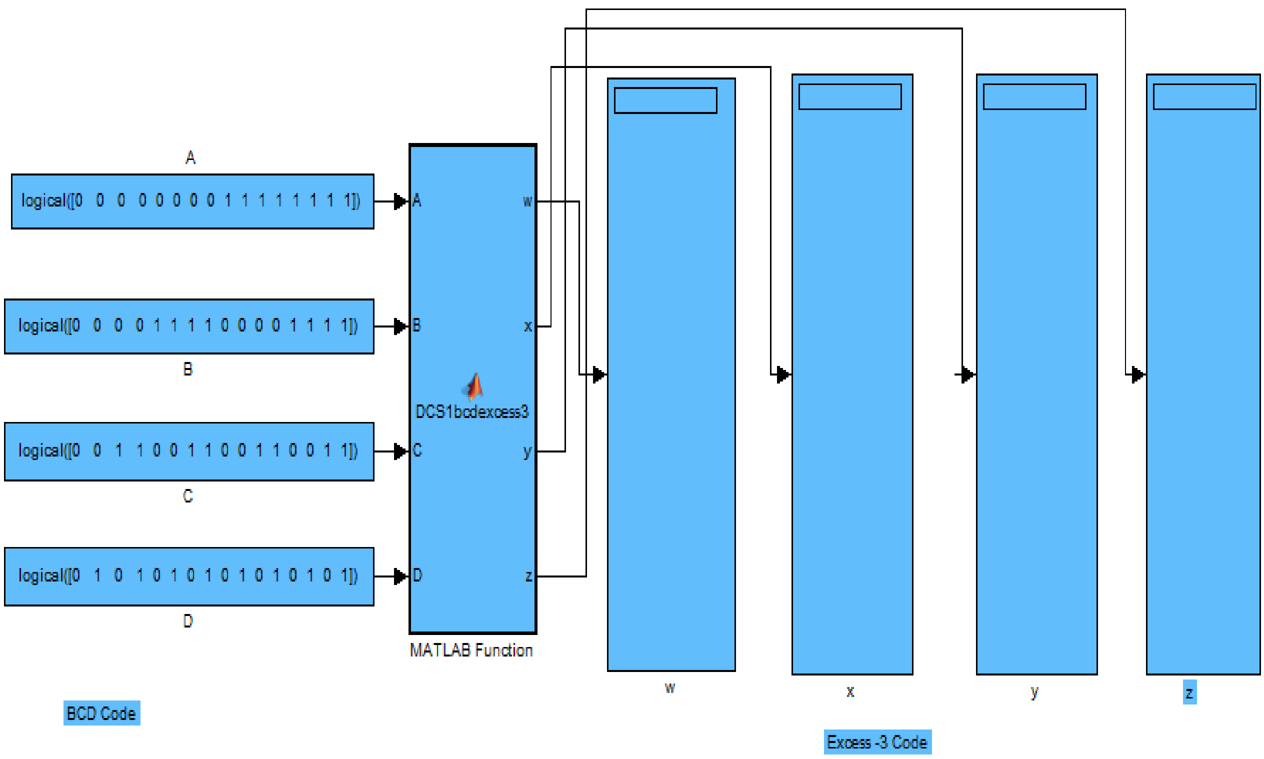

Fig.15. Realization of Binary Decimal Code to Excess-3 Code

```
Function [w, x, y, z] =DCS1bcdexcess3 (A, B, C, D)
% Usage:
%BCD to EXCESS-3 code conversion
%**********************************************************
% Define input variable of function DCS1bcdexcess3
%**********************************************************
% Inputs:
% A: Logical input
% B: Logical input
% C: Logical input
% D: Logical input
%**********************************************************
% Output variable of function DCS1bcdexcess3
%**********************************************************
%Output:
%BCD to Excess-3 code
For m=1:16                    % loop initialization
   z=~D;
   y= (C&D)| (~C&~D);
   x=~B& (C|D) |B&~ (C|D);
   w=A|B& (C|D);
End
```

Fig.16. MATLAB Script for BCD to Excess-3 code Convertor

V. RESULT AND DISCUSSION

In present proposed study, it is tried to foster the generic purpose, scientific computing, integrated development environment Matlab to digital electronics fellows.

VI. FUTURE SCOPE

It is well noticed from this study, simulation environment is more student centric learning to individual nascent. Searching or scientific computing can be easily implemented by using Matlab. Learning of Matlab provides multi-facet skill to diverse student in internet based fast growing technology oriented world.

REFERENCES

[1] Harold K and Randal A, Simulation of dynamical system with MATLAB and Simulink, 2nd edition, CRC Press, 2011.
[2] Ibrahim D, Engineering Simulation with MATLAB: improving teaching and learning effectiveness, Procedia Computer Science, 2011, 3: 853-858.
[3] Feng P, Mingxiu L, Dingyu X, Dali C, Jianjiang C, Application of MATLAB in teaching reform and cultivation of innovation talents in universities, IEEE proceedings of 2nd International Workshop on Education Technology and Computer Science,2010, 700-703.
[4] Tahir H, Pareja T F, MATLAB package and science subjects for undergraduate studies, International Journal for Cross-Disciplinary Subjects in Education, 2010, 1(1): 38-42.
[5] Osowski S, Simulink as an advanced tool for analysis of dynamical electrical systems, Computational Problems of Electrical Engineering, 2011, 1(1):51-60.
[6] J Ryan A Tarcini, Preparing students to accelerate innovation through simulation based engineering and sciences, IEEE 2nd International Workshop proceedings on Education Technology and Computer Science, 2010, 2: 700-703.
[7] http://www.mathworks.com
[8] Attia, J.O. Electronics and circuit analysis using MATLAB, CRC Press, 2004.
[9] Dabney, James B. And Harman, Thomas L, Mastering Simulink, Pearson Education, 2004.
[10] Karris, S. T. Introduction to simulink with engineering applications, Orchard publications, 2006.
[11] Karris, S. T., Signal and system using MATLAB, Orchard Publications.
[12] Kalechman, M, Practical MATLAB applications for engineers, CRC Press, 2009.
[13] Palamids A. and Veloni A., Signals and systems laboratory with MATLAB, CRC Press, 2010.
[14] M. Blaho, M. Foltin, P. Fodrek and P.Poliacik, "Preparing advanced MATLAB users", WSEAS Transaction on Advances in Engineering Education, vol.7, no.7, 2010, pp.234-243.
[15] S. Jain, Modeling and simulation using MATLAB-Simulink, Wiley India, 2011.
[16] William J Palm, A concise introduction to MATLAB, Tata McGraw-Hill, 2012.
[17] Blaho, M, Foltin, M, Fodrek, P. and Murgas, J. "Education of Future Advanced MATLAB Users, MATLAB-A", Fundamental Tool for Scientific Computing and Engineering Applications-Volume 3, InTech, 2012.
[18] Bober W and Stevens A, Numerical and Analytical Methods with MATLAB for Electrical Engineers, CRC Press, 2013.
[19] R.P. Jain, Digital Electronic, Tata McGraw hill Education, 4rd edition, (2010)
[20] Digital Integrated Electronics, Herbert Taub, Donald Schiling, Tata McGraw hill, 6th reprint, 2010
[21] Morris Mano, Digital Logic and computer Design, PHI Learning Private Limited New Delhi, 1979
[22] Charles Mclean and Swee Leong Manufacturing and Modeling National Institute of Standards and Technology Gaitherburg, MD 20899

[23] Studying the Role of Simulation in Developing Students' Statistical Reasoning, Andrew Zieffer, and Joan B. Garfield.
[24] Simulation Modeling I organizational and Management Research J. Richard Harrison Zhiang LIN University of Texas of Dallas
[25] Simulation: An Enabling Technology in software Engineering Alan M. Christe
[26] John F. Wakerley, Digital Design Principles and practices, Pearson, 3rd edition, 2001
[27] Elbert Pual Malvinoand Donal Leach, Digital Principles and Applications, 4th Edition, Tata McGraw Hill Publishing Company Ltd., New Delhi, India, 1991.
[28] Yap Wing Fen, Luq Man Al Hakim Mohammad Sabri, Integration of LabVIEW for Novel Interactive Learning Courseware on Digital Electronics, IJIER, Vol.2 , Issue 11, pp 156-163, 2014
[29] J.A. Fran Cisco, A.D. Castro, and B. Christian, Course on Digital Electronics Oriented Describing System in VHDL, IEEE Transactions on Industrial Electronics, Vol. 50, pp 3308-3316, 2010
[30] Teyana, Interactive Web based Laboratories in Digital Electronics, International Conference on Education and e-learning Innovation, vol.1, pp 1-5, 2012
[31] Suresh Kumar A, S.Muruganand, International Journal of Innovative Research in computer and communication engineering ,vol.2,Issue 4,PP 3748 -3755,2014.
[32] Jena, Manaswini Dehury, Anita Mohanty, Sushree Sourvi Kar, ALU Design and Its Perfomance Analysis in LabVIEW, ITSI Transaction on Electrical and Electronics Engineering (ITSI-TEEE) Vol. 2, Issue -5,6 pp 2320-8945, 2014
[33] S. Salivahanan and S. Arvazhagan, "Digital Circuits and Design", 3rd ed. Vikas publishing, India, pp. 168-169. 63, pp. 271-350.
[34] M. Morris Mano, "Register transfer and Microoperations," in Computer System Architecture, 3rd ed. Pearson, India, pp. 106-118.
[35] Professor Barry Paton, "Fundamentals of Digital Electronics" using LABVIEW, March 1998 edition, Part Number 321948A-01.
[36]
[37] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
[38] J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
[39] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.
[40] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in Proc. ECOC'00, 2000, paper 11.3.4, p. 109.
[41] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sep. 16, 1997.
[42] (2007) The IEEE website. [Online]. Available: http://www.ieee.org/
[43] M. Shell. (2007) IEEE transaction webpage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/IEEEtran/
[44] FLEX Chip Signal Processor (MC68175/D), Motorola, 1996.
[45] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
[46] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
[47] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
[48] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997.

AUTHOR PROFILE

Mittarpal is working as Assistant Professor in Department of Electronics and Communication Engineering, Bhagat Phool Singh Women University, Sonipat Haryana, India. His area of specialization includes Nano Science and Technology, Electromagnetic Wave Theory, Antenna Wave Propagation, Electronic Device and Circuits, Digital Circuit Systems, Advance Digital Communication, Advance Digital Signal Processing, Design and Simulation, Matlab Programming, Simulink Graphical programming, PV Module Simulation.

Sunita Rani was born in Haryana on October 14, 1989. She received her B.Tech. degree in Electronics & Communication Engineering from Ch. Devi lal University, Sirsa, Haryana in the year 2011. She received her Master in technology (M.Tech.) in VLSI Design from YMCA University of Science &Technology, Faridabad, Haryana in 2014. Presently she is working in the Department of Electronics and Communication Engineering in Bhagat Phool Singh Mahila Vishwavidyalya, Khanpur-Kalan, Sonepat, Haryana, India. Her area of interest is design VLSI chip and she also works for Quantum Dot Cellular Automata technology for VLSI design.

Naresh Kumar was born in Haryana, India, on August 15, 1985. He received his Polytechnic Diploma in Information Technology from Board of Technical Education, Haryana in 2004. He obtained his B.Tech in Electronics & Communication Engineering from Punjab Technical University, Punjab in 2008. He obtained his Master in Technology (M.Tech.) in Electrical Engineering. Specialization in Electronics from YMCA University of Science & Technology, Faridabad, Haryana in 2011. He worked in Aravali college of Engineering & Management, Faridabad, Haryana from 2011 to 2014. From August 2014 to till date, he is working in SGT University, Gurugram, Haryana. His research interests includes VLSI circuit design, develop embedded system with various microcontrollers etc.