Arid Zone Journal of Engineering, Technology and Environment. August, 2014; Vol. 9:9-16

Print ISSN: 1596-2490, Electronic ISSN: 2545-5818
www.azojete.com.ng

DESIGN OF A 4-BIT MAGNITUDE COMPARATOR USING SIMULINK

Dibal P.Y.
(Department of Computer Engineering, University of Maiduguri, Maiduguri, Nigeria)
e-mail address: yoksa77@gmail.com

Abstract

In this paper, the design of a 4-bit magnitude comparator using Simulink is presented. An overview of a magnitude comparator is carried out in the first section, in terms of its application in engineering. The second section presents the methodology of the design; thus, a closer look at the magnitude comparator is achieved by looking at its block diagram, its truth table, and the equations which characterize the truth table. The second section also presents the Simulink modelling environment, with a snapshot of its interface and functions. The implementation of the design, using Simulink, is also presented in the second section. The design was achieved by splitting the design of the circuit into three stages. The third section of the paper presents the results and discussion. The results were obtained by simulating the magnitude comparator circuit with three sets of data. In the first set of data, data A was equal to data B. In the second set of data, data A was greater than data B. In the third set of data, data A was less than data B. The simulation produced the expected results in the case of each data set, thus verifying the accuracy of the design. The final part of the paper presents the conclusion arrived at, at the end of the design.

Keywords: Bit, simulink, comparator, electronics

1. Introduction

A comparator is an important arithmetic component in a digital circuit. This is because sorting (done by a comparator) is one of the most important problems in computer engineering/science (Geetanji *et al*., 2011). In communication and computing systems, many important processes require the sorting of data in the areas of multiprocessing and parallel computing (Geetanji *et al*., 2011).

Another application of the magnitude comparator is in the vending machine, in which a magnitude comparator is used to determine if the correct coin is inserted to make for the dispensing of the correct item (Tinder, 2000). The comparison of two numbers in digital systems is an arithmetic operation used to determine if one number is greater than, equal to or less than the other number (Anjuli and Satyajit, 2013). The magnitude comparator used for this purpose is a combinational circuit which compares two numbers, A and B, and determines their relative magnitude.

The magnitude comparator comes in the form of an Integrated Circuit (IC) form. The IC can be based on TTL (Transistor Transistor Logic) technology or it can be based on CMOS (Complementary Metal Oxide Semiconductor) technology (Fairchild Semiconductor, 1999).

It is the objective of this study to use Simulink to design the 4-bit magnitude comparator; which is a far better way of achieving the design than traditional methods like schematic capture. This is because Simulink allows us to focus on the modelling directly.

2. Methodology

2.1 The 4-Bit magnitude comparator

The block diagram of the circuit for comparing two n-bit numbers is shown in Figure 1 below:

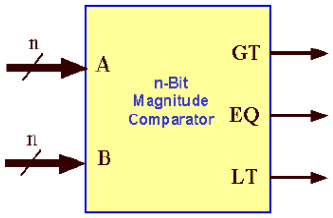

Figure 1: An n-bit magnitude comparator block

The circuit compares two n-bit binary numbers A and B, and gives three outputs:

GT (A > B), EQ (A = B), and LT (A < B)

To compare two numbers, the circuit must have ***2n*** inputs and 2^{2n} entries in the truth table (Anjuli and Satyajit, 2013). Thus, for 4-bit comparator, the inputs will be ***2x4 = 8*** inputs and the entries in the truth table will be $2^{2x4} = 2^8 = 256$. Due to space constraint, only a section of the entries of the truth table is presented below:

Table 1: Truth table for 4-bit

INPUTS								OUTPUTS		
A3	A2	A1	A0	B3	B2	B1	B0	A > B	A = B	A < B
0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	1	0	0	1
0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	1
0	0	0	0	0	1	1	0	0	0	1
0	0	0	0	0	1	1	1	0	0	1
0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	1	0	0	1	0	0	1
0	0	0	0	1	0	1	0	0	0	1
0	0	0	0	1	0	1	1	0	0	1
0	0	0	0	1	1	0	0	0	0	1
0	0	0	0	1	1	0	1	0	0	1
0	0	0	0	1	1	1	0	0	0	1
0	0	0	0	1	1	1	1	0	0	1
0	0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	0	0	1	0	1	0
0	0	0	1	0	0	1	0	0	0	1
0	0	0	1	0	0	1	1	0	0	1
⋮				⋮			⋮			⋮
1	1	1	1	1	1	1	1	0	1	0

As can be seen in Table 1, the 4-bit comparator compares the two numbers, each having 4-bits (A_3 to A_0 and B_3 to B_0).

2.2 The 4-Bit magnitude comparator design equations

The Boolean expressions (Anil, 2007) representing the conditions shown in the truth table (Table 1) are:

$$X = x_3.x_2.x_1.x_0 \quad where \quad x_i = A_i.B_i + \bar{A}_i.\bar{B}_i \qquad (1)$$

$$Y = A_3.\bar{B}_3 + x_3.A_2.\bar{B}_2 + x_3.x_2.A_1.\bar{B}_1 + x_3.x_2.x_1.A_0.\bar{B}_0 \qquad (2)$$

$$Z = \bar{A}_3.B_3 + x_3.\bar{A}_2.B_2 + x_3.x_2.\bar{A}_1.B_1 + x_3.x_2.x_1.\bar{A}_0.B_0 \qquad (3)$$

where ***X, Y, and Z*** are the three variables which represent the conditions ***A=B, A>B, and A<B*** respectively. Equation 1 is explained by Table 2 below:

Table 2: Truth table for Equation1

Bit Value	Condition
x3=1	A3=B3
x2=1	A2=B2
x1=1	A1=B1
x0=1	A0=B0

ANDing x_3, x_2, x_1, *and* x_0 ensures ***X*** produces a '1' when x_3, x_2, x_1, *and* x_0 are all in the logic ***'1'*** state.

The variable ***Y*** in equation 2 will be ***'1'*** if ***A>B*** during the comparison operation. Along the same line, if ***A<B***, the variable ***Z*** will be ***'1'***. As an example if we have the data shown in Table 3 below:

Table 3: Sample input for equation 2

A3	A2	A1	A0
1	1	1	1

B3	B2	B1	B0
1	1	0	1

It can be seen that at ***A>B***, because ***A1>B1***. Hence substituting this into equation 2 at $x_3.x_2.A_1.\bar{B}_1$ stage of the computation results in ***'1'***, hence ***Y=1*** because the operation of an OR gate is being performed and when one of the inputs is ***'1'***, the output of the gate is ***'1'*** irrespective of what the other inputs may be.

2.3 Simulink environment

The Simulink environment makes it possible to design and develop different types of engineering models, from simple models to advances models, as long as such models can be characterized by equations. In the Simulink environment, an engineer can carry out modelling through graphical procedure. The ability to carry out graphical programming makes Simulink

superior to its companion, MATLAB (Karris, 2006). This is because the design engineer can focus his attention more on conceptualization of the design, than the syntax.

The snapshot in Figure 2 below shows the Simulink environment.

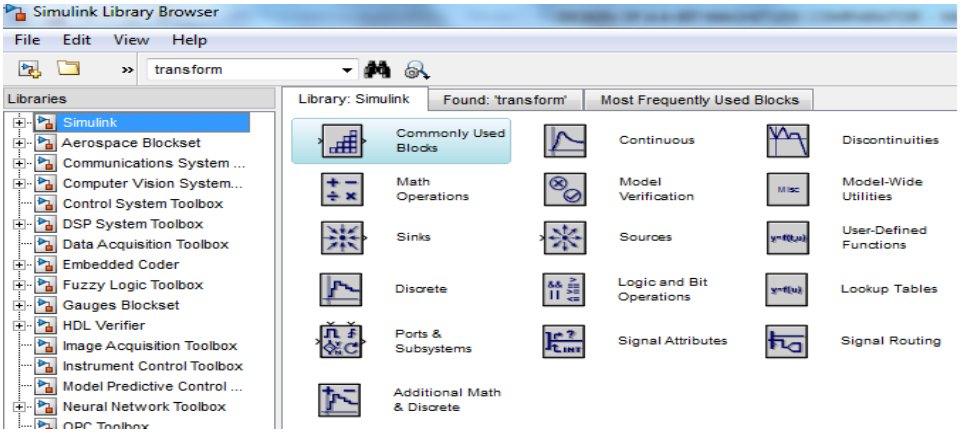

Figure 2: Snap Shot of Simulink Environment

As can be seen from the snapshot in Figure 2, the Simulink environment has many libraries, each of which is loaded with a lot of functions for effective engineering design and simulation.

To use the design in Simulink, we click on the New Model button, as shown Figure 3.

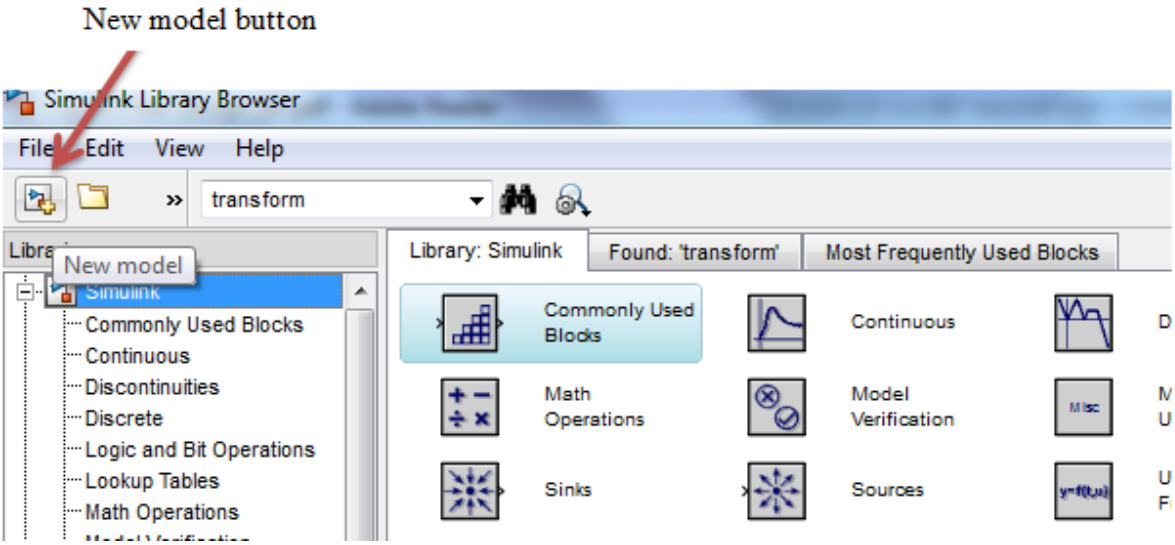

Figure 3: Starting a new model in Simulink

Clicking the New model button will launch a design entry interface (Figure 4), in which we can design the magnitude comparator.

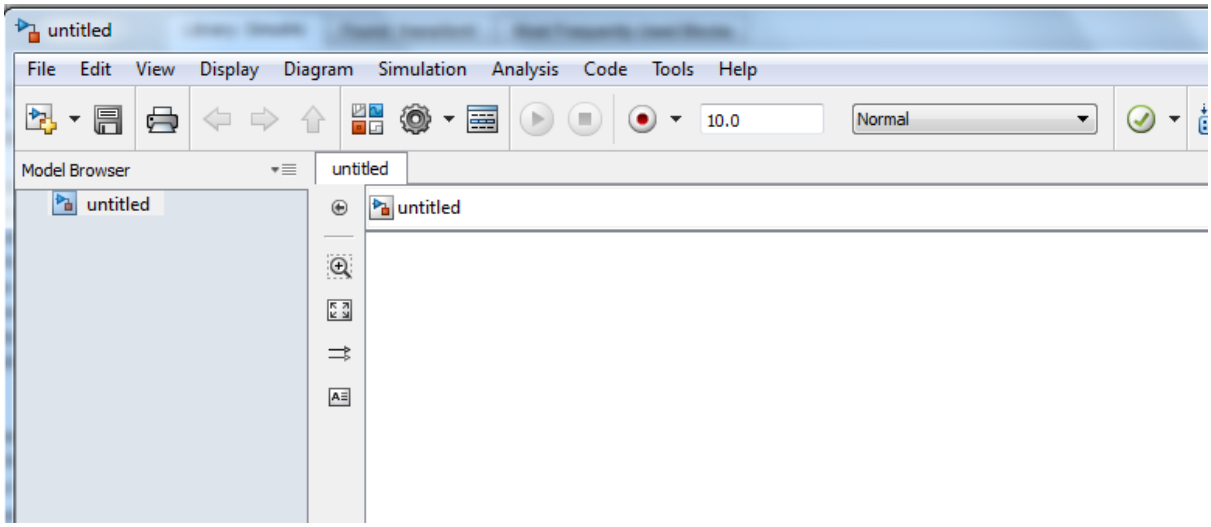

Figure 4: Design entry interface

2.4 Design implementation

Employing Simulink and using equations 1, 2, and 3, the design of the 4-bit magnitude comparator is presented in Figure 5 below:

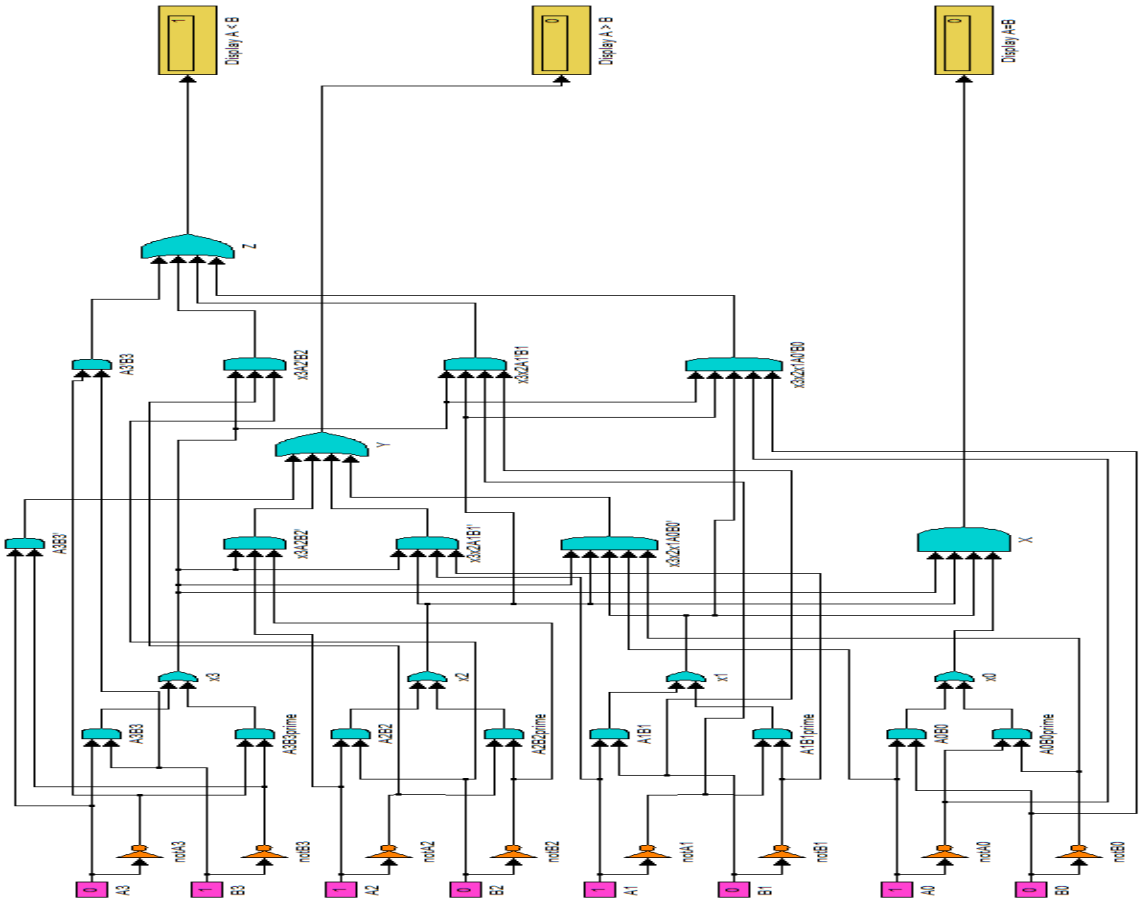

Figure 5: A 4-bit magnitude comparator using Simulink

The design shown in Figure 5 is divided into three parts: the input stage, the combination stage, and output stage. The input stage produces the input signals (4-bit wide), with their corresponding complements, as shown in Figure 6 below:

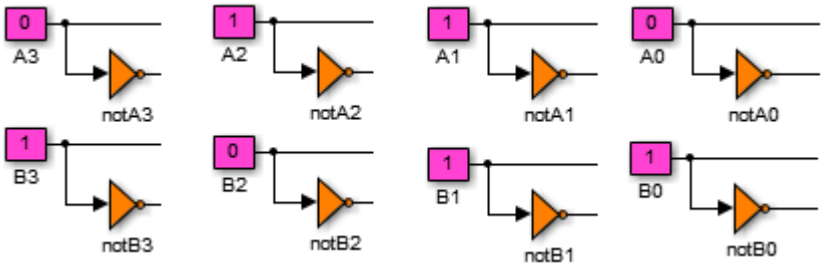

Figure 6: Input stage of magnitude comparator circuit

The combination stage is made of OR and AND gates; this is because the equations used in the design of the 4-bit magnitude comparator involve logical OR and AND operations (Figure 7).

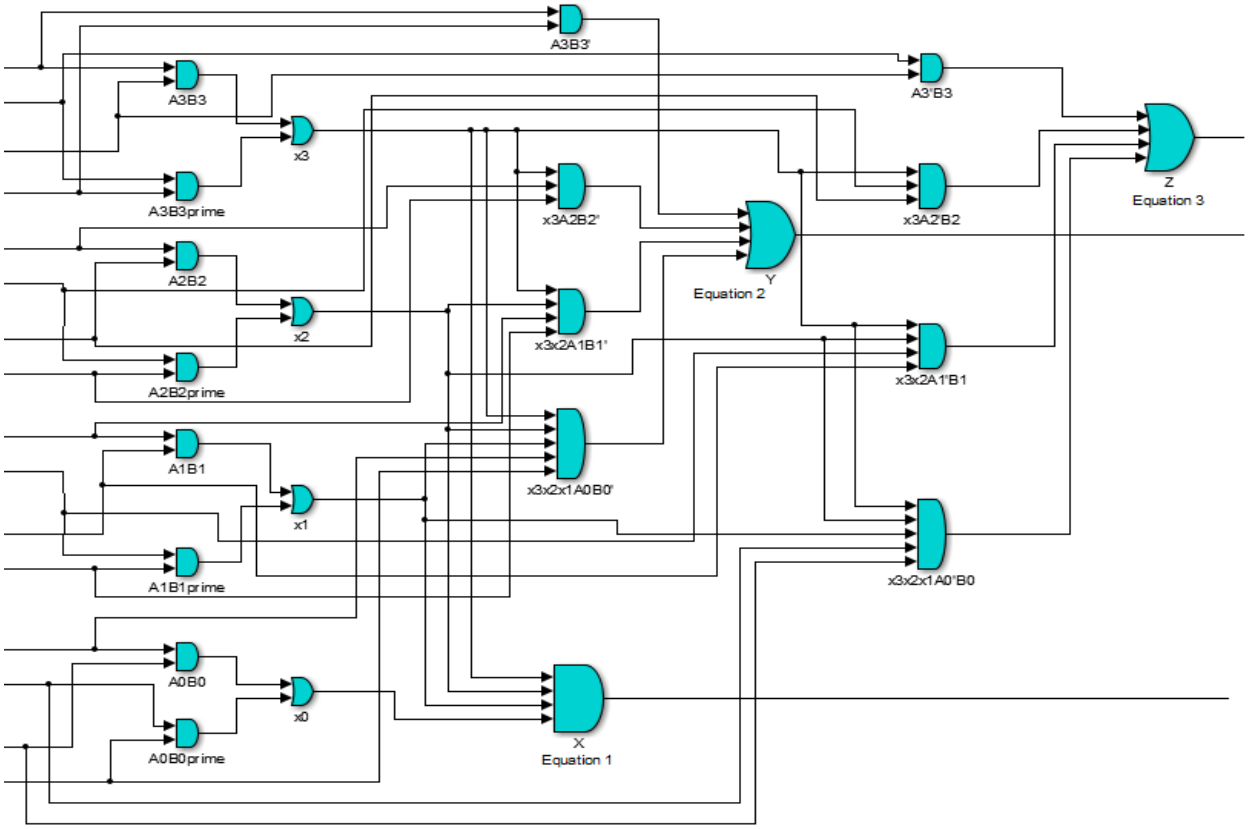

Figure 7: Combination stage of magnitude comparator circuit

The third and final stage of the comparator circuit is the output stage. This stage, using the Display function block of Simulink displays the simulation results of the comparator circuit (Figure 8).

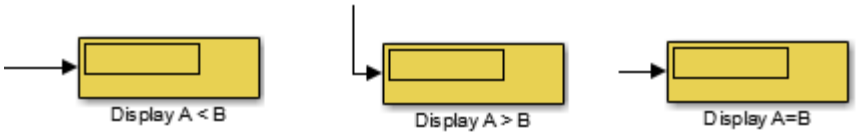

Figure 8: Output stage of magnitude comparator

3. Results and discussion

To test and verify the accuracy of the 4-bit magnitude comparator, we simulate it with three set of values, for when ***A=B***, ***A>B***, and ***A<B.*** The three set of data are presented in Table 4 a, b and c below:

Table 4: Three sets of data used in simulating the 4-bit magnitude comparator

a : Case when A = B

A = B								
A3	A2	A1	A0		B3	B2	B1	B0
1	0	1	1		1	0	1	1

b : Case when A > B

A > B								
A3	A2	A1	A0		B3	B2	B1	B0
1	0	0	0		0	1	1	1

c : Case when A < B

A < B								
A3	A2	A1	A0		B3	B2	B1	B0
0	1	1	0		1	0	1	1

Figure 9 shows that the output of the magnitude comparator is asserted for the display ***A=B*** because the two values being compared are, equal in magnitude.

when ***A=B:***

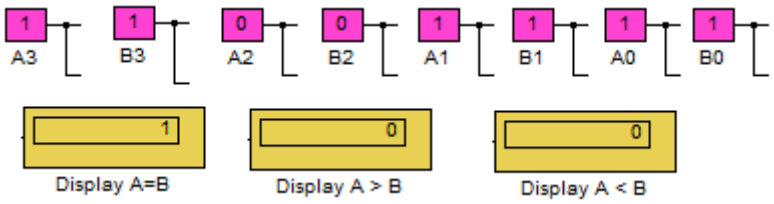

Figure 9: Comparator output when ***A=B***

Figure 10 shows the output of the magnitude comparator being asserted for the display ***A>B*** because for the two values being compared ***A*** is greater than ***B.***

when ***A>B:***

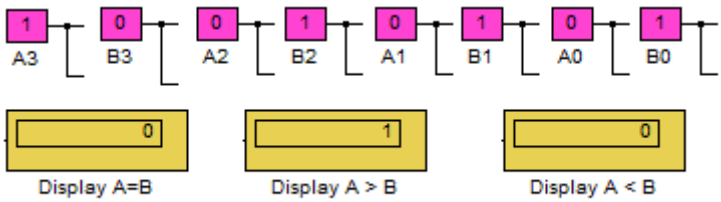

Figure 10: Comparator output when ***A>B***

Figure 11 shows the output of the magnitude comparator being asserted for the display ***A<B*** because for the two values being compared ***A*** is less than ***B.***

when ***A<B:***

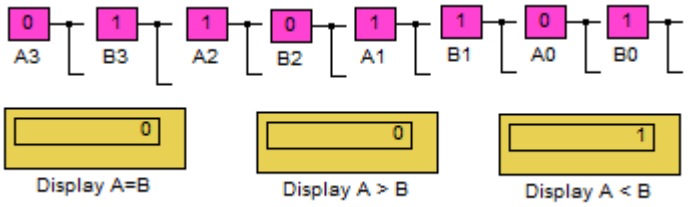

Figure 11: Comparator output when ***A<B***

4. Conclusion

Based on equations 1, 2 and 3, a 4-bit magnitude comparator was effectively and correctly designed using Simulink. The simulation of the comparator was carried out successfully and the results obtained were as expected. Therefore, the objective of designing the 4-bit magnitude comparator has been achieved.

References

Anjuli, S. and Satjajit, A. 2013a. 2-bit magnitude comparator design using different logic styles. International Journal of Engineering Science Invention, **2**(1): 13-24.

Anjuli, S. and Satjajit, A. 2013b. High-speed 64-bit CMOS binary comparator. Innovative Systems Design and Engineering, **4**(2): 45-58.

Anil, KM. 2007. Digital Electronics, Principles, Devices and Application. West Sussex: Willey & Sons.

Fairchild Semiconductor 1999. Magnitude Comparator Data Sheet. New York: Fairchild Semiconductor Co. Inc.

Geetanjali, SU. and Yogesh, M. 2011. A low-power 8-bit magnitude comparator with small transistor count using hybrid PTL/CMOS logic. IJCEM International Journal of Computational Engineering and Management, **12**: 110-115.

Karris, ST. 2006. Introduction to Simulink with Engineering Applications. New York: Orchard Publications.

Tinder, RF. 2000. Engineering Digital Design. 2nd ed. San Diego: Academic Press.