Journal of the Nigerian Association of Mathematical Physics
***Volume* 25 (November, 2013), pp 339 – 352**
© J. of NAMP

Design Implementation and Simulation of a 4-Bit Decimal Adder using Parallel Binary Adders and BCD Encoders

Galandanci G. S. M. and Gana S. M.

Physics Department,
Bayero University, P.M.B 3011 Kano.

Abstract

In this work a 4-bit decimal adder was designed, implemented and simulated. The decimal addition is carried out using Binary Coded Decimal (BCD) addition and a 10-line to 4-line encoder was upgraded to 15-lines to 4-lines and was used to convert each decimal digit to its corresponding BCD code. A 4-bit parallel adder with fast carry (74HC283), and additional circuitry of AND gate and OR gates were interconnected to carry out the BCD addition. A BCD to seven segment display converter (DCD Hex) was used to convert and displayed the generated sum in BCD to decimal. The designed system was implemented and simulated on circuit simulation software (National instrument multisim version 11). The system is tested and found to be capable of performing addition of any two 4-bit decimal numbers.

***Keywords*: tuberculosis, homotopy analysis method, nonlinear equations, mathematical model, epidemics**

1.0 Introduction

The most basic function and fundamental uses of digital systems is Numerical computation. In this era of technological advancement almost all computers, simple and sophisticated electronic gadget have a built-in capability of performing various arithmetic computations most of which are carried out using the binary (base 2) number system. Therefore there is need to design, implement and later construct digital combinational elements that will be able to perform binary arithmetic functions to execute the desired operations of addition, subtraction, multiplication and division of these numerical values.These numerical values or numbers are usually represented either in binary or in decimal through a binary code. In most arithmetic systems, users like to give and view or displaythese values in decimal form. However, if the inputted number is in decimal, it is converted to binaryequivalent and the corresponding arithmeticoperations are done internally in the system by means of these converted binary codes. Hence, the purpose of this research is to provide the basic information on howto formulatesystematic design and simulation of simple arithmetic circuit or system that will be able to perform such functions. This system can be used to perform binary addition of two decimal digits say X and Y from 0 to 15respectively applied via two different design keyboard encoders. The keyboard X and Y contain 15 different switches; each of which represent a decimal digit; that are connected to a priority encoder responsible for the conversion of those decimal numbers into their corresponding Binary Coded Decimal (BCD) equivalent. When one of the switches is close a BCD outputthat is equivalent to a single selected decimal input is generated, and the Decimal Coded Decimal (DCD) Hex display displays the corresponding decimal digit.

2.0 Theoretical Background

2.1 BCD or 8421BCD Code

Binary-coded decimal (BCD) also referred to as 8421 code is used to represent each digit of a decimal number as a 4-bit binary number. It is used to combine the features of decimal base system (base 10) and binary base number system (base 2) to explore a wider function of performing digital mathematical operations [1]. The designation of 8421 indicates the basis of binary weight of 4-bit binary numbers in descending power of 2, starting from 2^0 (1) as the weight of the least significant bit (LSB) to the next higher order bits 2^1 (2) , 2^2 (4), up to the Most significant bit (MSB) weight of 2^3(8). In practice, there are

Corresponding author: ***Galandanci G. S. M.,*** E-mail: gsmgalad@yahoo.com, Tel.: +2348037260380 sixteen (16) 4-bit binary numbers (0000 through 1111) but only ten of these numbers 0 to 9 (0000 through 1001) are used and are referred to as the valid BCD codes[2]. Hence the binary codes 10 to 15 (1010 through 1111) are never being used in BCD code and are referred to as the invalid BCD codes. TABLE1 gives the valid and the invalid codes and their decimal equivalents.

TABLE 1: Valid and invalid BCD codes

Decimal digit	Valid BCD (8421) codes	Decimal digit	In-Valid BCD (8421) codes
0	0000	10	1010
1	0001	11	1011
2	0010	12	1100
3	0011	13	1101
4	0100	14	1110
5	0101	15	1111
6	0110		
7	0111		
8	1000		
9	1001		

Moreover, BCD addition of a group of 4-bit BCD numbers is accomplished by the same rules governing the mathematical operation performed in the addition of a group of 4-bit binary numbers. It should be noted that since only ten (0 to 9) out of 16 4-bit binary codes can be represented in the BCD code, then it is necessary to consider the possibility of getting a sum that is a 4-bit binary number greater than nine (9) - the six invalid BCD codes [3]. If the sum of the 4-bit is equal to or less than nine (9), it is simply a valid BCD number and we do not have problem. However, if the sum is greater than nine (9) or a carry is generated out of the 4-bit group resulting in an invalid BCD state then a binary number 6 (0110) is added to the sum in order to skip the six (6) invalid states and return back to the valid BCD code. Also if the addition of the number six (6) results in a carry, the carry is added to the next 4-bit group [2].

2.1 DecimalEncoder

A Digital Encoder also commonly known as **decimal Encoder**is a combinational circuit that converts active level information from its inputlines to a unique number of output lines; it takes *allof*its data inputs one at a time and then converts them into a single encoded output[1].The operation of the encoder is such that exactly one of the input lines should have a one (1) while the remaining input lines should have a zero (0)and vice versa for an Active HIGH input or an Active LOW input depending on the type of integrated circuit used or application requirements. The output lines generate a coded binary output from a single active numerical input that corresponds to the input value [4]. Thus, it can be said that an encoder, is a multi-input combinational logic circuit that converts the logic level say "1" data at its inputthat correspond to a single decimal value into an equivalent binary code at its output [3].

Furthermore, it produces several outputs of 2-bit, 3-bit, 4-bit or 8-bit codes that depends upon the number of desired data input lines required. Figure 1 shows a nine input Encoder (one for a single decimal digit) with four binary outputs and TABLE2 shows the decimal inputs and binary outputs of an encoder which corresponds to a valid BCD/8421 codes.

Table 2: Decimal inputs and binary outputs of an encoder

Decimal digit	Binary equivalents Number			
	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

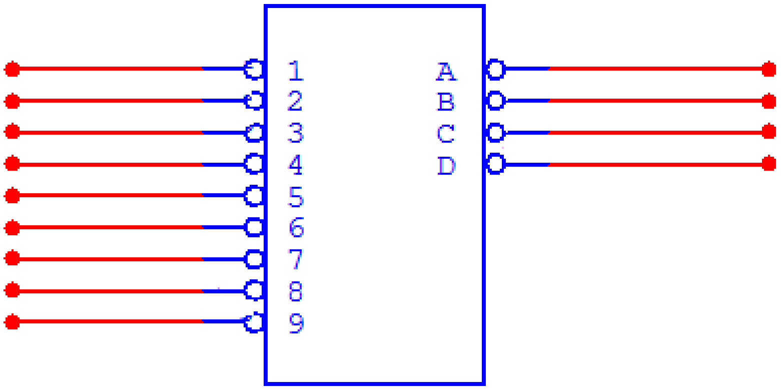


Figure 1: A nine input Encoder with four binary outputs

2.3 Priority Encoder

The binary encoder has the limitation that only one input can be active at any given time. If two inputs are active simultaneously, the output produces an undefined combination. For instance, input line 1 and 4 are depressed simultaneously, the output of the encoder will produce an output value that is neither1 nor 4. Hence, in order to resolve this problem the encoder circuits must assign a priority to each of the input lines so that when several input lines are activated or depressed, the encoder outputs the index value of the input line with the highest priority. For example, if 2 and 7 are depressed simultaneously the output will be 0111 (decimal 7). This modified encoder is known as a **priority encoder**[2]**.** 74HC147 is a typicalpriority encoder integrated circuit.

2.4 Parallel Binary Addition

In digital systems, parallel addition of binary numbers is performed by two or more full-adders connected together. A full-adder is a combinational circuit that forms the arithmetic sum of three input bits to produce a resulting sum (∑) and a carry out (C_{out}) from its output. Two of the inputs say A and B represent two significant bits to be added (operands) and the other input represent carry in (C_{in}) from the previous lower significant bit position if available otherwise the C_{in} input is grounded when there is no input to the least significant bit position [3]. TABLE3; shows the operation of a full adder.

Table 3: Operation of a full adder

Decimal digit	Inputs			Outputs	
	A	B	C_{in}	SUM	C_{out}
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

The numbers of full-adders required for the addition of any group of binary numbers is determined by how many bits can be used to represent each number. For instance, to add two 2-bit binary numbers two adders are needed; for 3-bit three adders are needed; for 4-bit four adders are needed continuously [5]. The integrated circuit 74HC283 is an example of a 4-bit parallel adder; internally it contains a group of four full-adders. Figure 2 shows an example of parallel full adder for 2-bit addition.

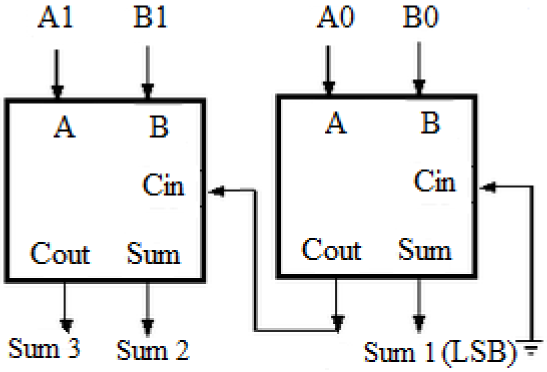


Figure 2: A parallel full adder for 2-bit addition.

3.0 Design and Implementation

To have a clear understating of the basic concept for this design the system is divided into two sections.

1. Decimal to Binary encoding section
2. Overflow detector circuit section
3. Binary to Decimal codes addition section

These three sections when connected together form the 4-bit decimal adder using parallel binary adders and BCD encoders. They can be implemented, tested and simulated individually using National instrument multisim version 11.0.

3.1 Decimal to Binary Encoding Section

The decimal to binary encoding section employed two different fifteen digits decimal keyboard denoted X and Y that can be used to input two different decimal numbers from 0 through15. The keyboard is connected to a priority encoder that converts the said decimal numbers into their corresponding binary coded decimal (BCD) equivalent.The 74HC147 priority encoder can only encode numbers between 0 to 9; 0 when all lines are in active and 1 to 9 when an in active level is applied to one of its input lines. However our main objective is to add two 4-bit binary numbers from 0000(0) up to 1111(15) in using our system when decimal equivalent of these binary numbers are inserted via a keyboard.Thus there is a need to design another encoder that will be able to encode the remaining 4-bit number from 10 (1010) to 15(1111) and will work synchronously with the 74HC147 integrated circuit to have a complete 0 to 15 encoding circuit for proper 4-bit addition. To design these encoder we consider TABLE 4 so as to determine the relationship between each of the 4-bits codes in order to develop proper combination logic circuit to perform the encoding function.

TABLE 4:Decimal Digit 10 to 15

Decimal Digit	D(MSB)	C	B	A(LSB)
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

From the TABLE 4 it can be seen that the most significant it (MSB) for **D** is always a 1 for the entire decimal digit from 10 to 15, hence an expression can be obtained for the decimal digit from an OR function.

$$D = 10 + 11 + 12 + 13 + 14 + 15$$

Similarly Bit **C** is always a 1 for decimal digit 12, 13, 14 and 15 thus its OR function expression can be express as;

$$C = 12 + 13 + 14 + 15$$

Bit **B** is always a 1 for decimal digit 10, 11, 14 and 15 and the OR function expression can be express as follows;

$$B = 10 + 11 + 14 + 15$$

Finally for the least significant bit (LSB) **A** is always a 1 for decimal digit 11, 13, and 15 and can be express as follows;

$$A = 11 + 13 + 15$$

The logic expressions develop are than implemented as shown in figure 3 for encoding the required decimal digits by ORing the decimal digit input lines to their appropriate binary equivalents. Moreover, since the 74HC147 integrated circuit has an Active-LOW input an inverter is connected at all the input lines of the design encoder to operate with Active-LOW level.

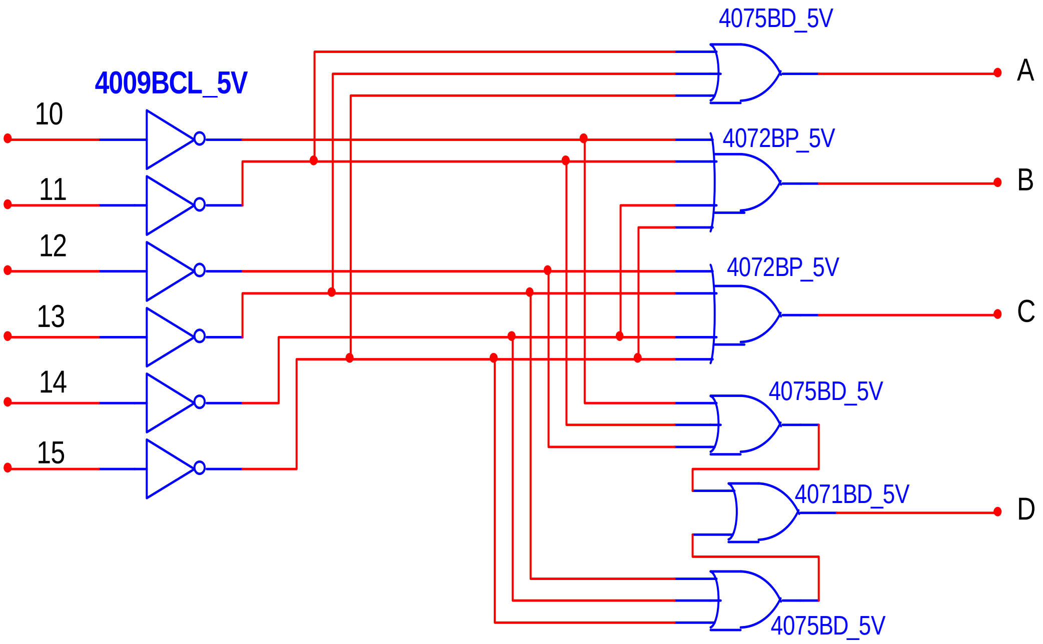


Figure 3: Design Encoder for Decimal Digit 10 to 15

The design encoder and the 74HC147 IC are then connected together for complete 4-bit encoding function as shown in Figure 4. When one of the keys from keyboard X is pressed for a giving decimal digit, the decimal digit is processed by the 74HC147 priority encoder and the result is displayed as a corresponding BCD code and then inputted to one of the input lines of the parallel adder. Each of the switches of the keyboard is connected to a Pull-Up resistor; the main essence of this is to ensure that the line connected to the encoder is kept HIGH when the switch is not depressed since the priority encoder (74HC147) has an active-LOW input for decimal digit 0 through 9 and an active-LOW BCD output.The size for a pull-up resistor depends on the size of the output’s load and leakage current through the integrate circuit when it is LOW. A good choose the resistor is 10KΩ because it is not too small to allow excessive current to flow through the input lines and not too large to cause an excessive voltage drop when the input lines are LOW [6] for these reason one end of each switch is connected to the ground so that whenever a switch is depressed the line connected to the corresponding encoder input becomes LOW and this result in each outputs of the encoder to be either HIGH or LOW for the resulting BCD number of the depressed decimal numbers as desired. When all the switches are not connected or depressed the BCD output represent a zero.Figure 3shows the keyboard encoder. It can be seen that the input lines are held HIGH by the pull-up resistors connected to +5 V [7].

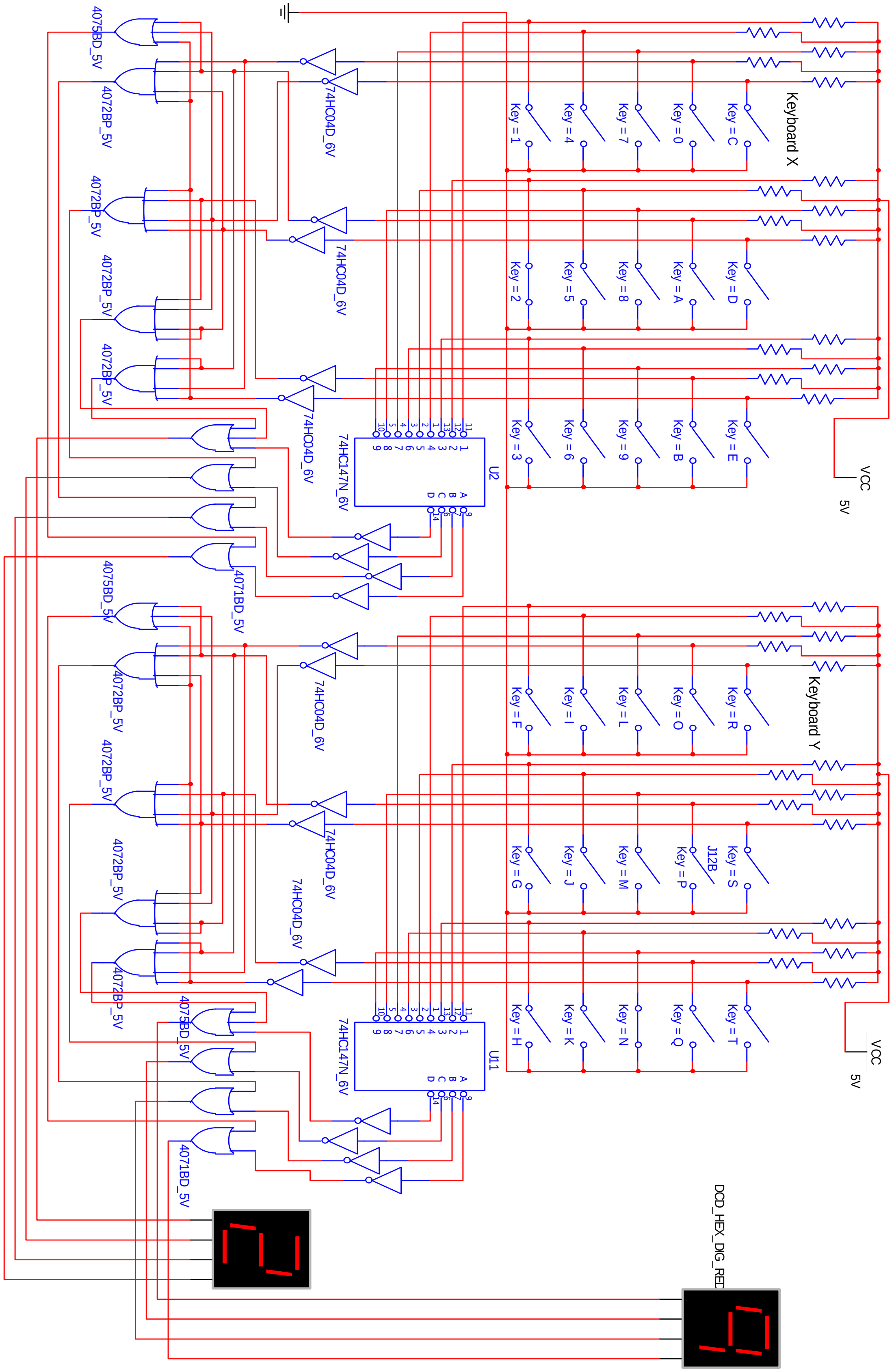


Figure 4: Keyboard Encoder

When one of the switches is close a BCD output is generated that is equivalent to a single selected decimal input, and the DCD displays the corresponding decimal digit. The 74HS147 decimal-to-BCD (10-line-to-4-line) priority encoder IC has active-LOW outputs and hence produces a negative logic. To get a negative true logic an inverter 4004 is connected. The choice to use positive or negative true logic depends on application of your design.For example, if decimal number 2 is depressed from the keyboard say X the output of the encoder produce its BCD equivalent from the most significant bit to the least significant bit as 0100 and displays it in the DCD display and then inputted to one of the input of the parallel adder for the BCD addition. Similarly, the same process applied to keyboard Ywhen decimal digit nine (9) is depressed as shown in Figure 4.

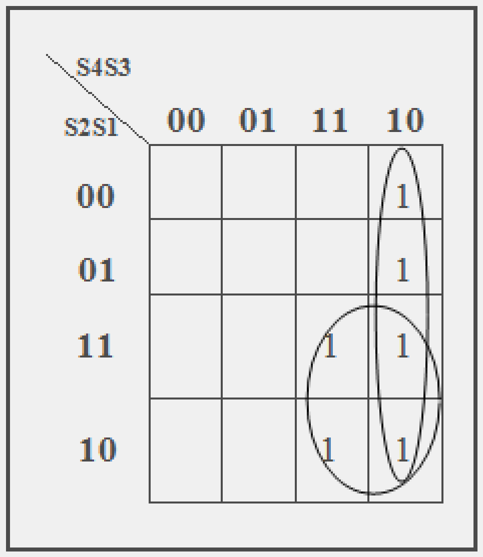
3.2 Overflow Detector Circuit

In this design, both the keyboard encoders X and Y can only input a maximum decimal digit fifteen (15) and thus the maximum computed value or summed value will be thirty (30); 15 + 15. As previously mentioned when the sum of the two digits is less than or equal to nine (9) than the ordinary 4-bit adder is sufficient for the addition of the two numbers. However, if the sum is greater than 9 or a carry out (C_{out}) in one of the 4-bit group is generated resulting in a sum greater than 15 (maximum 4-bit digit), then the sum produce, extend to a 5-bit binary digit which does not exceed the maximum number that the circuit can compute if carry out (C_{out}) output is utilize, thusa correction must be added to skip the invalid state. These detected problems arose the need to design a circuit that will be able to detect any overflow and have the capability of performing the correct BCD addition in order to have a comprehensive BCD-based summation outputs [7]. To design the overflow detector circuit let us consider TABLE 4 that gives the case where the sum of two 4-bit numbers is greater than 9.

Table 5: Sum of two 4-bit numbers is greater than 9

Decimal digit	C_{out}	Sum 4 (S_4)	Sum 3 (S_3)	Sum 2 (S_2)	Sum 1 (S_1)
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1
20	1	0	1	0	0
21	1	0	1	0	1
22	1	0	1	1	0
23	1	0	1	1	1
24	1	1	0	0	0
25	1	1	0	0	1
26	1	1	0	1	0
27	1	1	0	1	1
28	1	1	1	0	0
29	1	1	1	0	1
30	1	1	1	1	0

From TABLE 5; it can be seen that whenever the sum is greater than 15 resulting to 5-bit digit,carry out isalways equal to 1 ($C_{out} = 1$). Hence there is overflow at $C_{out} = 1$. However, when the sum is between 10 (1010) to 15 (1111) there is no carry out ($C_{out} = 0$) yet an overflow occurs, thus, since 10 to 15 contain 4-variables excluding the carry; a standard sum of product (SOP) plotted on a 4-variable karnaugh map (k-map) can be used to determine the correct and simplified Boolean expression for this overflow using the assumption that when an overflow occurs the expected output value should always be one (1) [8]. Considering only S_4 to S_1 for digit 10 to 15 the K-map is plotted on Figure 5.



S2S1 \ S4S3	00	01	11	10
00				1
01				1
11			1	1
10			1	1

Figure 5: Karnaugh map for deduction of overflow detector boolean function.

The expression to design the complete overflow detector circuit can be expressed by summing C_{out} (since $C_{out} = 1$ when there is a carry) and the minimum generated Boolean expression from the k-map. Hence the expression is given below;

$$overflow \ = C_{out} + \ s_4(s_3 + s_2)$$

Figure 6 shows the implemented circuit connected to a 4-bit parallel adder for proper BCD addition.

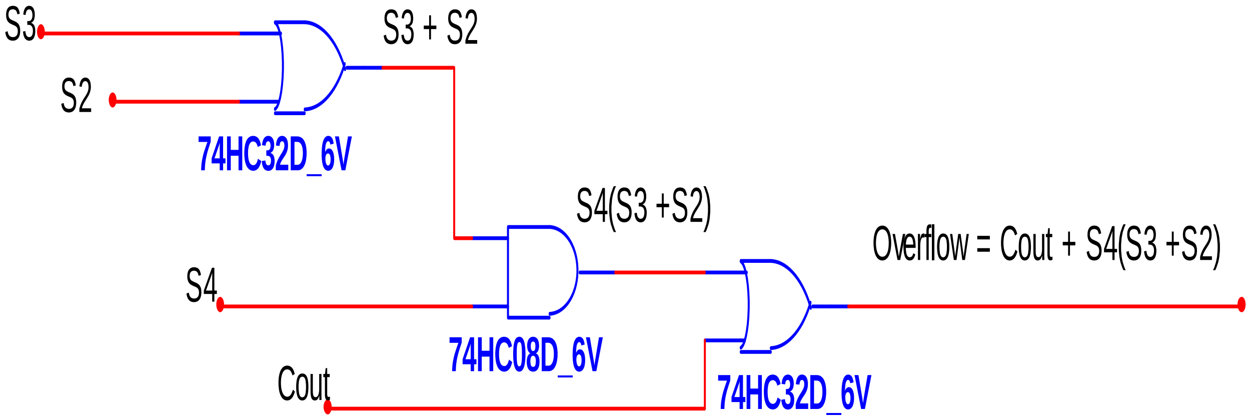


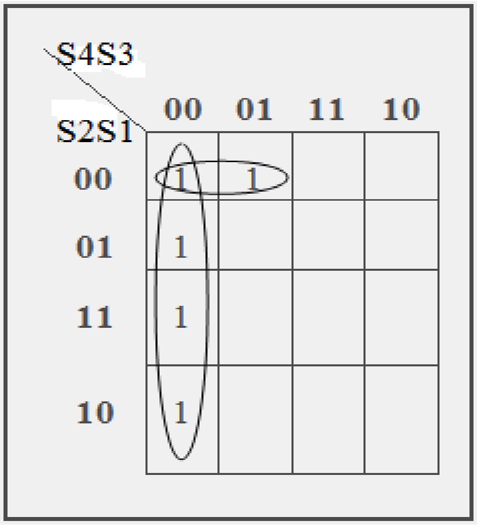
Figure 6: Overflow Detector Circuit

Furthermore, in some cases six (0110) must be added twice or thrice through different detectors to skip the invalid BCD states. There are special cases where when six (0110) is added a valid BCD code is produced, however another six (0110) must be added to obtain the correct result of the summed digit. These special numbers are 26, 27, 28, 29 and 30 and If six (0110) is added to them the resulting sum produce are valid BCD code 0(0000), 1(0001), 2(0010), 3(0011) and 4(0100) respectively as shown in TABLE6.

Table 6: Sum of numbers that produce a valid BCD code when six(0110) is added.

Decimal digit	C_{out}	Sum 4 (S_4)	Sum 3 (S_3)	Sum 2 (S_2)	Sum 1 (S_1)	Sum when six(0110) is added			
						S4	S3	S2	S1
26	1	1	0	1	0	0	0	0	0
27	1	1	0	1	1	0	0	0	1
28	1	1	1	0	0	0	0	1	0
29	1	1	1	0	1	0	0	1	1
30	1	1	1	1	0	0	1	0	0

It can be noticed from TABLE6that for proper display of the correct digit six must be added; hence the need to design a detector for this purpose arises. The detector can be developed by taking thestandard sum of product (SOP) plotted on a 4-variable karnaugh map (k-map) of the number given in TABLE 6 when six (0110) is added to determine the correct and simplified Boolean expression for this overflow.



S2S1 \ S4S3	00	01	11	10
00	1	1		
01	1			
11	1			
10	1			

Figure 7: K-map for deduction of overflow detector circuit

The minimum expression obtain from the k-map is given below;

$$Overflow = \overline{S4S3} + \overline{S4}\,S3\,\overline{S2S1}$$

Figure 8 shows the implemented circuit connected to a 4-bit parallel adder for proper BCD addition.

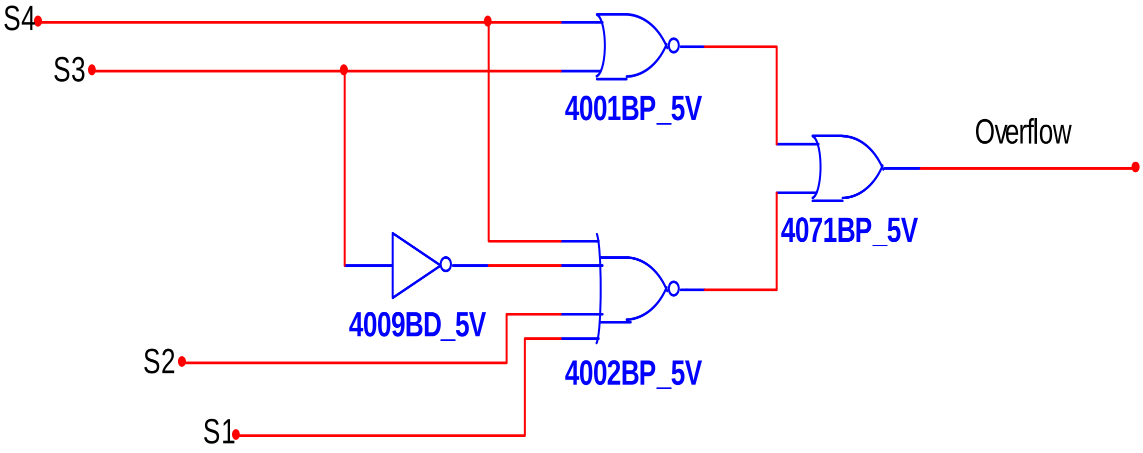


Figure 8: Overflow Detector Circuit when valid BCD is produce

3.3 BCD Addition Section

The parallel adder 74HC283 IC compose of two inputs A and B each of which contain an individual four line input for corresponding 4-bit binary number addition from most to least significant bit $A_{4,}A_{3,}A_{2,}A_1$and $B_{4,}B_{3,}B_{2,}B_{1,}$respectively. However, in order to accommodate and enable the addition of binary number 6 (0110) from the overflow detector circuit whenever an invalid BCD number is summed, two of 74HC283 parallel adders are used [7]. The parallel adder is responsible for the addition of these binary numbers from 0 through 9 when the keyboard is pressed. For example if decimal number 3 is pressed from keyboard X and decimal number 6 is pressed from keyboard Y, the encoder convert them to their BCD equivalent and then applied it to the parallel adder inputs A ($A_{4,}A_{3,}A_{2,}A_{1,}$) and B ($B_{4,}B_{3,}B_{2,}B_{1,}$) as 0011 (binary 3) for input line A and 0110 (binary 5) for input line B accordingly as shown in figure9.

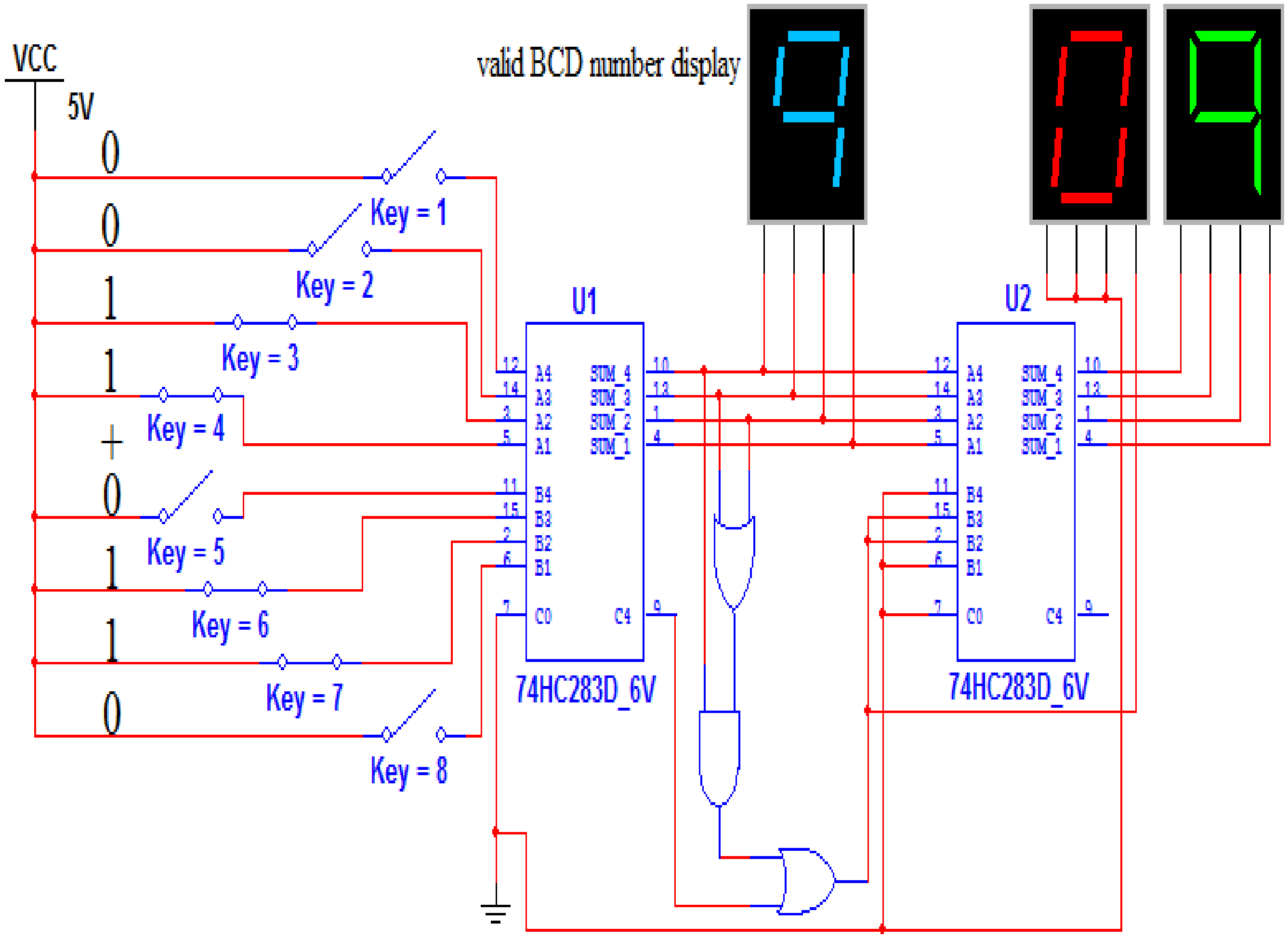


Figure 9: Binary to Decimal Coded Addition circuits for valid BCD sum

The addition 0011 + 0110 = 1001 (binary 9) is carried out internally by the adder and the computed result is displayed by the DCD display in its decimal value.Since the sum is a valid BCD code and there is no carry out, the overflow detector circuit is inhibited thus produce 0 in the input lines B_3 and B_2 of parallel adder U2; that is binary 0 (0000) is added to the sum instead of binary 6 (0110) which does not affect the computed sum. If the computed sum result in an invalid BCD number, then the overflow detector circuit produces 1 in the input lines B_3 and B_2 of parallel adder U2 enabling the addition of binary six (0110) to skip the invalid BCD states. Figure10shows an invalid BCD sum display in decimal from the addition of binary 9 (1001) and binary 6 (0110). The sum is displayed in hexadecimal before the proper addition of binary six (0110) from the overflow detector circuit [2].

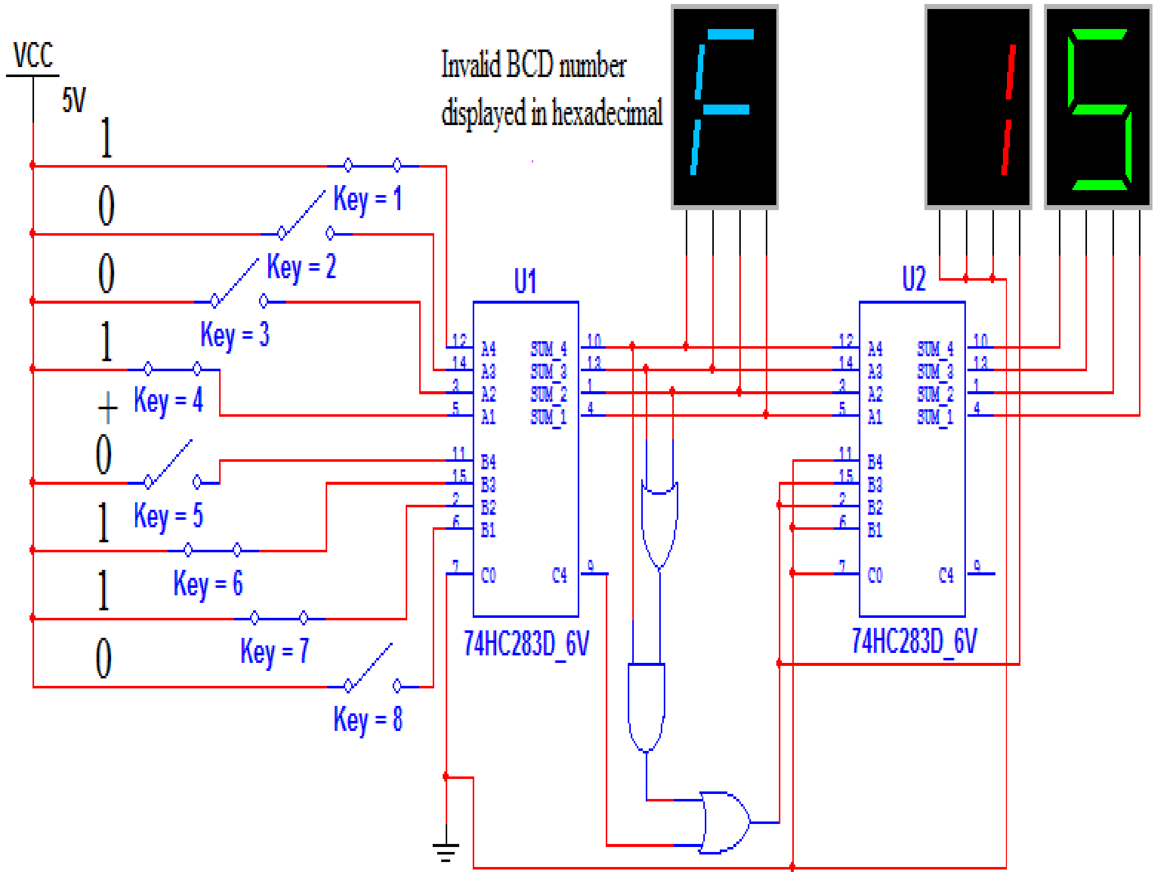


Figure 10: Binary to Decimal Coded Addition circuits for in-valid BCD sum

3.4 General Circuit Diagram

The general circuit of the 4-bit decimal adder using parallel binary adders and BCD encoders is designed to work synchronously with the three sections to have an appropriate and accurate BCD addition. Figure 11 shows the general circuit diagram. For the general operation of the complete circuit it can be notice that when the sum produce is less than 10 (1010) from 0 (0000) to 9 (1001) which is a valid BCD code all the overflow detectors from the 2nd and 3rd adders are inhibited and the result is displayed in the unit DCD HEX display as its decimal equivalent, similarly for the tenth display since all the inputs of the full adder are 0 it produces a sum and a carry of 0 and is displayed as 0 to complete the display of numbers ranging from 00 to 09. If the sum produce is greater than 9 (1001) or a carry out ($\mathbf{C}_{out}$) in one of the 4-bit groups is generated resulting in a sum greater than 15 (maximum 4-bit digit) that extent to a 5-bit binary digit, the 1st overflow detector from the 1st adder is activated for sum between 16 (10000) through to 19(10011). These produces an invalid BCD code and therefore binary six (0110) is added to the 2nd adder to skip the invalid state, the sum produce at the output of the 2nd adder is now a valid BCD code thus passes through all the successive adders to display the corresponding decimal digit while all their detectors are inactive. At the same time since the 1st detector is active input **A** of the full adder connected to it is now 1 while all the other inputs (**B** and $\mathbf{C}_{in}$) are 0, these produces a sum of 1 and a carry of 0 at its output and is displayed as 1 in the tenth decimal display to complete the display of numbers ranging from 10 to 19.

Also if the sum produce is still a 5-bit binary digit that extend from 20 (10100) to 29 (11101) the detector from the 1st adder becomes active and binary six (0110) is added through the 2nd adder. However, at the output of the 2nd adder an invalid BCD code is also produce and to skip the invalid state two detectors are connected at its output. The 1st detector enables the addition of binary six (0110) when an overflow occurs at the output of the 1st adder and an invalid BCD code is produce at the output of the 2nd adder after the addition of six (0110) from its input; and the 2nd detector enables the addition of binary six (0110) only when a carry out ($\mathbf{C}_{out}$) is produces from the 1st adder and a valid BCD code is produce from the output of the 2nd adder. If the sum produce is between the range of 20 (10100) to 25 (11001) the detectors from the 1st adder and the 1st detector from the 2nd adder are activated and hence input **A** and **B** of the full adder connected to them are all 1 while $\mathbf{C}_{in}$ is 0, these produced a sum of 0 and a carry of 1 from the output of the full adder to give a binary equivalent of two (10) which is now displayed as the corresponding decimal equivalent in the tenth decimal DCD HEX display. At the same time binary six (0110) is added to the 3rd adder to display the corresponding decimal digit in the unit display simultaneously to complete the display of numbers ranging from 20 to 25 while all the remaining successive detectors are inactive. Moreover, if the sum produce is between 26 (11010) to 29 (11101) the same process is repeated only that that the 2nd detector becomes active and 1st detector inactive thus only the **A** input and **B** input of the full adder are 1, and the corresponding results are displayed in the tenth and unit display respectively.

Finally, when the sum produce is 30(11110) an overflow occurs at the 1st detector hence a 1 is provided at the input **A** of the full adder while 6(0110) six is added to the 2nd adder simultaneously. Automatically the 2nd detector at the output of the 2nd adder becomes active (a valid BCD code is produce and detector from 1st adder is active) thereby providing a 1 at $\mathbf{C_{in}}$ input of the full adder and enabling the addition of 6(0110) to the 3rd adder. Similarly, at the output of the 3rd adder an invalid BCD code is produce thus enable the addition of 6(0110) to the 4th adder to skip the invalid state; these produce a carry at the output of the 4th adder which in essence provide a 1 at the **B** input of the full adder through the OR gate connected to it. With the resulting sums provided at the output of the 4th adder while all the full adder inputs are 1 that produce a sum and a carry of 1 to give a binary equivalent of three(11) , number 30 is displayed in decimal in the tenth and unit display of the DCD HEX display.

4.0 Result and Analysis

The complete circuit of the 4 bit decimal adder shown in Figure 11 was implemented and the simulation carried out using circuit simulator software produce by national instrument, Multisim version 11.0.The simulation of the 4-bit decimal adder was carried out by selecting some set of decimal numbers randomly and summed together to give either a valid or invalid BCD number.TABLE7 gives the sum of 10 different decimal numbers in BCD inputted via the keyboard encoder and the displayed decimal equivalent after the BCD addition is performed by the circuit.

Table 7: Simulation Analysis

Inputted Decimal digit		Computed sum in BCD	Displayed result in decimal	Remark
Keyboard X	Keyboard Y			
1	4	0101	5	Valid BCD
9	8	10001	17	In-valid BCD
3	6	1001	9	Valid BCD
7	5	1100	12	In-valid BCD
4	3	0111	7	Valid BCD
2	8	1010	10	In-valid BCD
9	9	10010	18	In-valid BCD
4	5	1001	9	Valid BCD
7	8	1111	15	In-valid BCD
5	7	1100	12	In-valid BCD

The analysis shows that even if the sum is greater than the valid BCD sum the result is displayed; this is due to the fact that the overflow detector enable the addition of six (0110) whenever its output is 1 for invalid BCD sum hence, the same 1 is displayed for proper display of the computed value.

5.0 Conclusion

In conclusion the design of 4-bit decimal adder using parallel binary adders and BCD encoders was successfully carried out using the appropriate combinational logic elements. The designed system was implemented and simulated using national instrument version 11.0; and thesimulation result shows that the system functions as desired, where any two 4-bit numbers from 0 to 15 can be added by BCD addition and display the result in decimal equivalent.

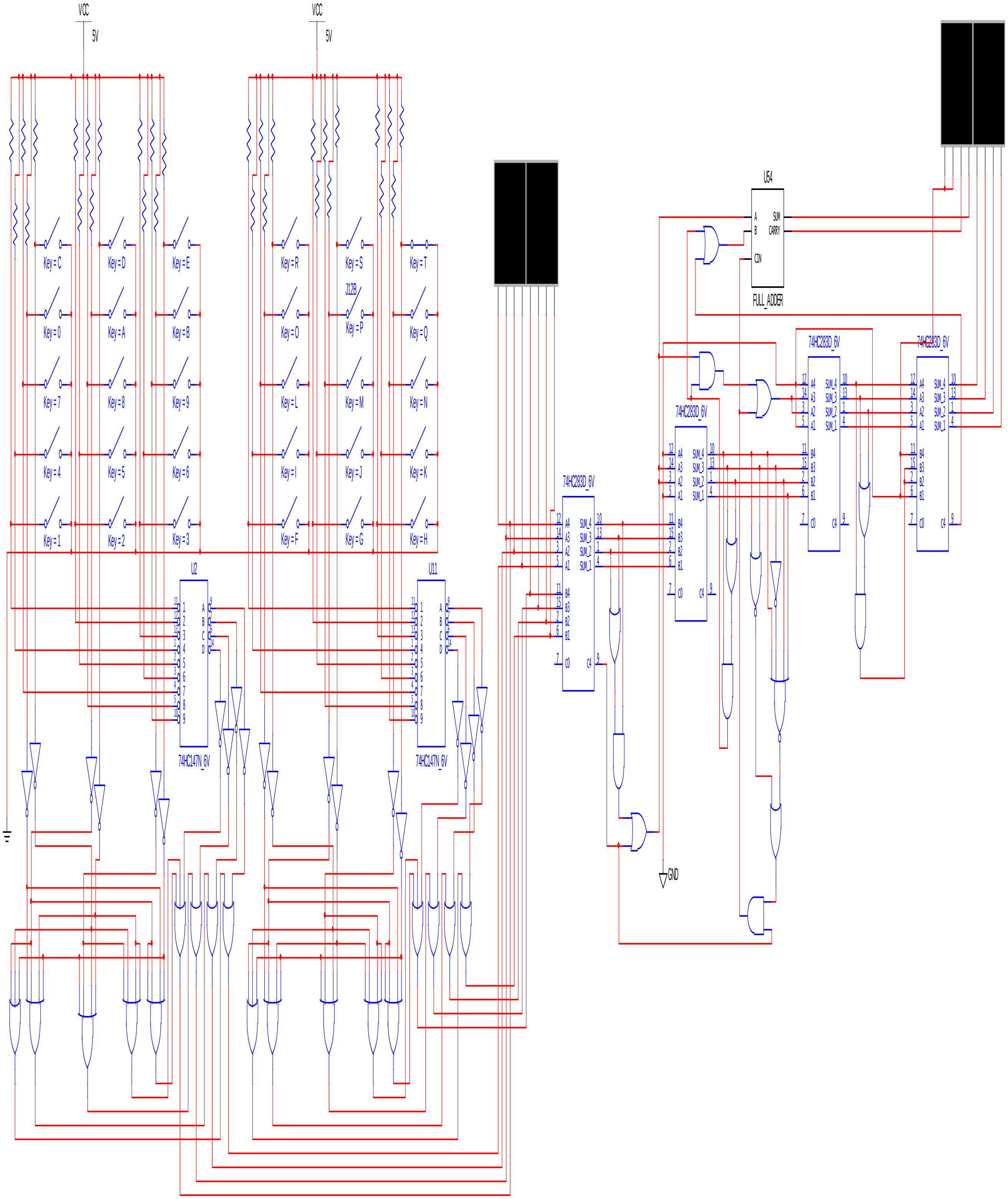


Figure 11: The general circuit diagram for a 4-bit decimal adder using parallel binary adders and BCD encoders.

References

[1] John, F.W.(1999), Digital Design Principles & Practices 3rd Edition, Prentice Hall, Upper Saddle River, New Jersey.

[2] Thomas, L.F.(2007), Digital Fundamentals 9th Edition, Pearson Prentice Hall, Pearson Education Singapore Pre. Ltd.

[3] Ronald, J. T.(2001) Digital Systems Principles and Applications 8th Edition, Prentice Hall, Upper Saddle River, New Jersey.

[4] Myke, P.(2005), Digital Electronics Demystified, McGraw-Hill Companies, Inc., New York.

[5] Mark. B(2003), Digital Design-A Comprehensive Guide to Digital Electronics and Computer System Architecture, McGraw-Hill Companies, Inc, New York..

[6] Barry, P, Fundamentals of Digital Electronics, National Instruments Corporation, Texas.

[7] Paul, S.(2000), Practical Electronics for Inventors, McGraw-Hill Companies, New York.

[8] Anil, K. M.(2007), Digital Electronics: Principles, Devices and Applications, John Wiley & Sons, Ltd.

[9] Itaketo,U.T.(2011),A Modified Logic Circuit of BCD Adder to Overcome the “Carry” Problems, The Pacific Journal of Science and Technology, Volume 12. Number 2.

[10] William, K,(2006), Digital Electronics with VHDL, Quartus II version, Prentice Hall, Upper Saddle River, New Jersey.