

Design of Digital Arithmetic Circuit Using Excess-3 Code & 9'S Complement Method

Dhrubojyoti Ghosh¹, Nilankar Bhanja²

¹(ECE, Abacus Institute of Engg. & Mgmt., Hooghly, West Bengal, India

²(ECE, Saroj Mohan Institute of Technology, Guptipara, West Bengal, India)

Abstract: This paper presents a technique to design an Excess-3 Arithmetic circuit capable of doing addition and subtraction operation using 9's complement method. The explored method of the circuit of 9's complement method along with the conventional adder-subtractor composite unit achieves the design to get perfect result. By this circuit we can add or subtract any two numbers with any sign in an efficient way by using Excess-3 arithmetic process without employing the human brain.

Keywords: Control inverter, Control sign input, Composite unit, Sign magnitude bit output, Full adders.

I. Introduction

The adder-subtractor composite unit has a vast application in digital arithmetic operations in digital technology. In digital world there are three sequential codes i.e. Binary code, BCD & Excess-3 code. Those codes can be used for arithmetic operations on digital system. This circuit is capable of adding and/or subtracting 2, two digit numbers resulting in a three digit number and an additional bit to represent whether the result is positive or negative. The operation of the circuit is totally depends on Excess-3 arithmetic process. The addition of two numbers can be achieved using several full adders. The subtraction of two numbers may be accomplished by taking 9's complement of the subtrahend and adding to the minuend. Control sign input in the present study controls the sign of the inputs as per requirement and thus can control the addition and subtraction using 9's complement method by implementing 1's complement method in the parallel full adder circuit. This circuit have inputs as X1X0 (as 2 Decimal number X), Y1Y0 (as 2 Decimal number Y), CX (as control input of X) & CY (as control input of Y) and capable of performing arithmetic operation such as +X+Y, +X-Y, -X+Y & -X-Y.

II. Theory

This circuit (Fig-1) have 2, two digit decimal number inputs as X (X1 X0) and Y (Y1 Y0). In X1, X0, Y1 & Y0 there are individual 0 to 9 total ten options (button) from where we can select any 2 two digit decimal number. Then we are using four Decimal to BCD encoder circuit to convert X, X, Y & Y into BCD codes (4bit). After that we using 4, four bit parallel adder by using full adders to convert BCD codes into Excess-3 codes by adding 0011 (3) with the BCD codes. Here the outputs of parallel adder 1 (S31 S21 S11 S01) represents the input Y0, Output of parallel adder 2 (S32 S22 S12 S02) represents the input X0, Output of parallel adder 3 (S33 S23 S13 S03) represents the input Y1 and Output of parallel adder 4 (S34 S24 S14 S04) represents the input X1 [1, 2, 4].

CA is the control sign input of X & CB is the control sign input of Y. Control sign inputs represents the sign of the inputs [5, 6].

$CA^* = CA * (CA \text{ XOR } CB)$; $CB^* = CB * (CA \text{ XOR } CB)$.

CA* is the complement controller of input X and CB* is the complement controller of input Y. When the sign of both the inputs are same then the circuit is behaving like normal adder, but when the sign of the inputs are different then the circuit is behaving like subtractor by employing complement and then addition process [2, 3].

Excess-3 is self-complementing code because we can perform 9's complement by employing 1's complement method. CA* & CB* are there to perform 1's complement of X & Y as per requirement by employing XOR gate as a control inverter.

Next we are using another two, 4bit parallel adder (parallel adder 5 & parallel 6) to add or subtract X0 with Y0 & X1 with Y1 [7, 8].

$K = \text{Cout}_6 * (CA \text{ XOR } CB)$ is used to add the carryout of MSB (parallel adder 6) with the LSB (parallel adder 5) whenever necessary as per the arithmetic rule of 1's complement method. After that the outputs of parallel adder 5 & 6 are fed into parallel adder 7 & 8, to convert the result into Excess-3 code as per the arithmetic rules of Excess-3 code.

$K^* = \text{Cout}_6 \wedge * (CA \text{ XOR } CB)$ is used to perform 1's complement of the result as per requirement. So there are 8 normal outputs to represent the result in Excess-3 code.

$C_{out} = C_{out6} * (CA \text{ XNOR } CB)$ is the extra output taken to represent the overflow output whenever necessary. By using parallel adder 9, C_{out} is converted into $C_{out3} C_{out2} C_{out1} C_{out0}$ (4 bit Carry output as Excess-3 code).

$SM = (CA * CB) + (C_{out6} \wedge (CA \text{ XOR } CB))$ is the sign magnitude output to represent the sign of the output [1, 2].

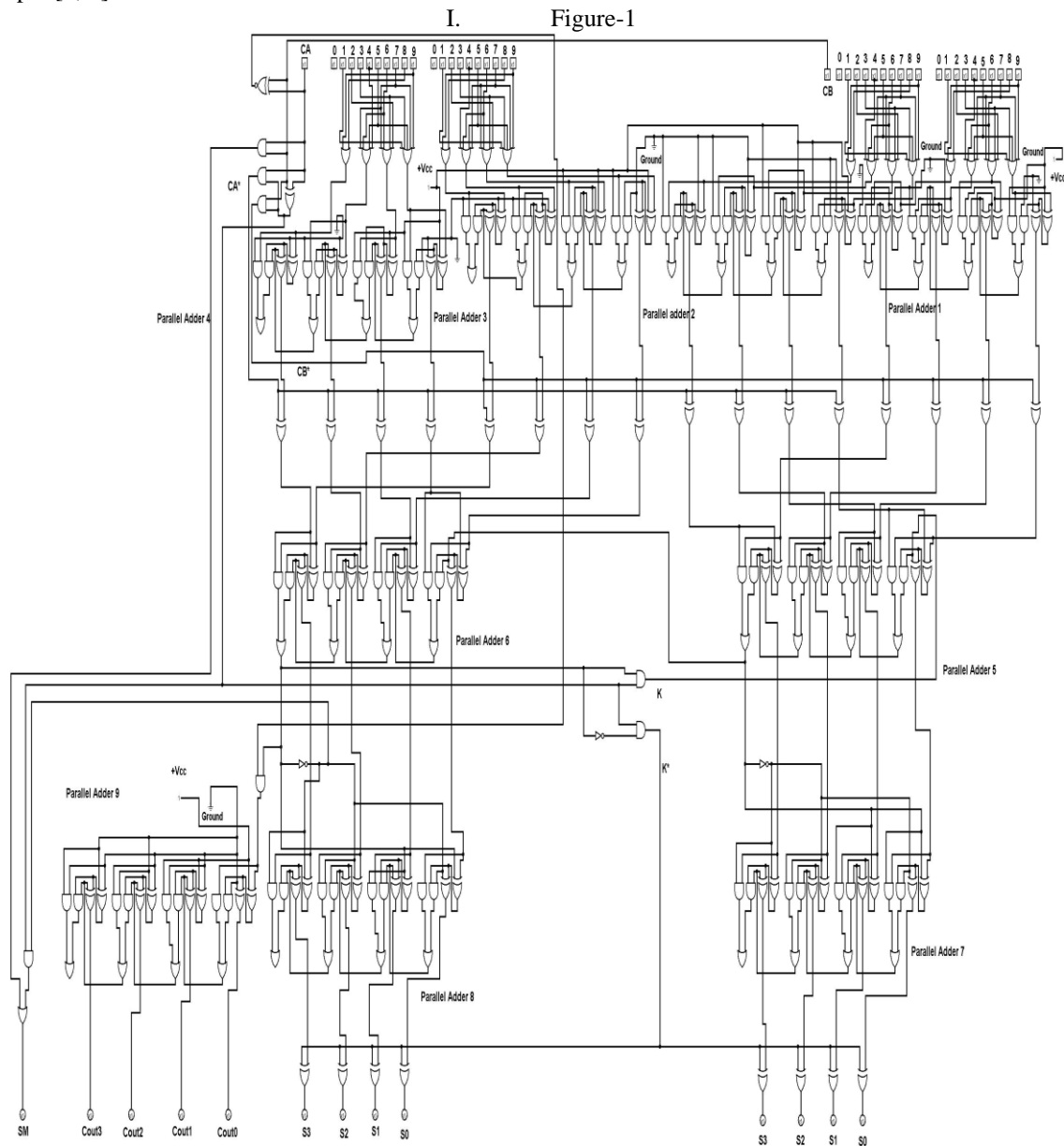


Table-1.a

CA	X								CB	Y							
	X1=9				X2=9					Y1=9				Y0=8			
0	11	11	00	00	11	11	00	00	0	11	11	00	00	11	00	11	11

Table-1.b

SM	Cout				S1				S0			
	Cout3	Cout2	Cout1	Cout0	S3	S2	S1	S0	S3	S2	S1	S0
00	00	11	00	00	11	11	00	00	11	00	11	00

Table-2.a

CA	X								CB	Y							
	X1=9				X2=9					Y1=9				Y0=8			
1	11	11	00	00	11	11	00	00	1	11	11	00	00	11	00	11	11

Table-2.b

SM	Cout				S1				S0			
	Cout3	Cout2	Cout1	Cout0	S3	S2	S1	S0	S3	S2	S1	S0
11	00	11	00	00	11	11	00	00	11	00	11	00

Table-3.a

CA	X								CB	Y							
	X1=1				X2=5					Y1=2				Y0=5			
1	00	11	00	00	11	00	00	00	0	00	11	00	11	11	00	00	00

Table-3.b

SM	Cout				S1				S0			
	Cout3	Cout2	Cout1	Cout0	S3	S2	S1	S0	S3	S2	S1	S0
00	00	00	11	11	00	11	00	00	00	00	11	11

Table-4.a

CA	X								CB	Y							
	X1=1				X2=5					Y1=2				Y0=5			
0	00	11	00	00	11	00	00	00	1	00	11	00	11	11	00	00	00

Table-4.b

SM	Cout				S1				S0			
	Cout3	Cout2	Cout1	Cout0	S3	S2	S1	S0	S3	S2	S1	S0
11	00	00	11	11	00	11	00	00	00	00	11	11

Table-5.a

CA	X								CB	Y							
	X1=3				X2=4					Y1=4				Y0=3			
0	00	01	11	00	00	11	11	11	1	00	11	11	11	00	01	11	00

Table-5.b

SM	Cout				S1				S0			
	Cout3	Cout2	Cout1	Cout0	S3	S2	S1	S0	S3	S2	S1	S0
11	00	00	11	11	00	00	11	11	11	11	00	00

Table-6.a

CA	X								CB	Y							
	X1=3				X2=4					Y1=4				Y0=3			
1	00	01	11	00	00	11	11	11	0	00	11	11	11	00	01	11	00

Table-6.b

SM	Cout				S1				S0			
	Cout3	Cout2	Cout1	Cout0	S3	S2	S1	S0	S3	S2	S1	S0
00	00	00	11	11	00	00	11	11	11	11	00	00

II. Result And Analysis

In Table-1, Table-1.a shows the 2bit decimal input X(X1 X0) as 99 and 2bit decimal input Y(Y1 Y0) as 98. The control sign input of X, CA is 0 Represent X as positive and the control sign input of Y, CB is 0 represent Y as positive. By the Decimal to BCD encoder & BCD to Excess-3 converter circuits, X (99) is converted into Excess-3 code as 1100 1100 and Y (98) is also converted into Excess-3 code 1100 1011. By processing through the circuit Table-1.b shows the final output as 0 0100 1100 1010. First MSB output SM =0 signifies the output as positive. Then four output Cout = 0100, next four output S1 = 1100 and last four output S0 = 1010 shows the Excess-3 result as + 4 12 10, that means the original answer is + 197. We know that, When X = +99, Y = + 98 then X + Y = + 197. So the simulation result satisfies the Original result.

In Table-2, Table-2.a shows the Input X(X1 X0) as -99 and Y (Y1 Y0) as -98. Table-2.b shows the final output as 1 0100 1100 1010 (Excess-3 form). So it signifies the result as -X -Y = - 197.

In Table-3, Table-3.a shows the Input X(X₁ X₀) as - 15 and Y (Y₁ Y₀) as + 25. Table-3.b shows the final output as 0 0011 0100 0011 (Excess-3 form). So it signifies the result as - X + Y = + 10.
In Table-4, Table-4.a shows the Input X(X₁ X₀) as + 15 and Y (Y₁ Y₀) as - 25. Table-4.b shows the final output as 1 0011 0100 0011 (Excess-3 form). So it signifies the result as + X - Y = - 10.
In Table-5, Table-5.a shows the Input X(X₁ X₀) as + 34 and Y (Y₁ Y₀) as - 43. Table-5.b shows the final output as 1 0011 0011 1100 (Excess-3 form). So it signifies the result as + X - Y = - 9.
In Table-6, Table-6.a shows the Input X(X₁ X₀) as - 34 and Y (Y₁ Y₀) as + 43. Table-6.b shows the final output as 0 0011 0011 1100. So it signifies the result as - X + Y = + 9.

III. Conclusion

The present circuit represent total 13 output bit, where MSB output bit SM represent the sign of the output, and the other 12 output as 3 Excess-3 code(4bit). First four bit in Excess-3 code as carry output and the last 8 bit as 2 Excess-3 code. In the input section we are using X(X₁ X₀) as 2 digit Decimal number and also Y (Y₁ Y₀) as 2 digit Decimal number. If X + Y or - X - Y exceeds the 2 digit range then this Carry out represent the exceed value. Otherwise carry out shows 0. The last 8 digits are used to represent the remaining result. Moreover, the present endeavour encompasses the use of the simple logic gates and costing to a minimum range enable to use in an economic manner in practical applicable field. In this circuit we can use 2 digit Decimal number as input, but we can also implement the same logic over higher digits of number also. In future I want to design an arithmetic circuit which is capable of using all three sequential codes in a single circuit.

Acknowledgements

We have been fortunate to get all support and cooperation from the management of Abacus Institute of Engineering & Management and Saroj Mohan Institute of Technology (Degree Division), Hooghly, West Bengal, India. We are very much thankful to our colleagues to support us in all respect.

References

- [1]. D. Ghosh and S. R. Saha, "Design of 4bit binary arithmetic circuit using 2's complement method", IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), 8(4), 2013, pp.39-42.
- [2]. D. Ghosh, K. K. Mandal and S. R. Saha, "Design of digital arithmetic circuit using BCD code & 9's complement method", International Journal of Electronics & Communication Technology (IJECT),4(4),2013,pp.115-117.
- [3]. D. Sangwan and M. K. Yadav, "Design and Implementation of Adder/Subtractor and Multiplication Units for Floating-Point Arithmetic", International journal of Electronics Engineering, 2(1), 2010, pp.197-203.
- [4]. D. Ghosh, "Design of 4-Bit Adder Subtractor Composite Unit Using 2's Complement Method"
http://www.academia.edu/4020322/DESIGN_OF_4BIT_ADDER_SUBTRACTOR_COMPOSITE_UNIT_USING_2S_COMPLEMENT_METHOD
- [5]. Goggle search, Digital adder circuit, http://minecraft.gamepedia.com/Tutorials/Advanced_redstone_circuit
- [6]. Google search,digital logic circuit design, <http://ozark.hendrix.edu/~burch/logisim/>
- [7]. U. Kulisch, "Advanced Arithmetic for the Digital
- [8]. Wikipedia,adder-subtractor composite unit, <http://en.wikipedia.org/wiki/Adder%E2%80%93subtractor>

Biographies



DHRUBOJYOTI GHOSH received the B.Tech & M.Tech Degree in Electronics and Communication Engineering from the University of W.B.U.T,Kolkata,West Bengal, in 2007 & 2009 respectively. Currently, He is an Assistant Professor of Electronics and Communication Engineering of Abacus Institute of Engineering & Management, West Bengal, India. His teaching and research areas include Digital Electronic circuits, Digital Communication Engineering. Dhrubojoyoti Ghosh (Assistant Professor) may be reached at dhrubo_ghosh@rediffmail.com



NILANKAR BHANJA awarded the B.Tech Degree in Electronics and Communication Engineering from the University of W.B.U.T, Kolkata, West Bengal, in 2005 & M .Tech Degree in Mechatronics from Bengal Engineering and Science University (BESU), currently known as IEST(Indian Institute of Engineering Science and Technology) Shibpur, Howrah, West Bengal. in 2009 . Currently, He is an Assistant Professor in the department of Electronics and Communication Engineering at Saroj Mohan Institute of Technology, West Bengal, India. His teaching and research areas include Digital Electronic circuits, Digital Communication Engineering, Mechatronics and Robotics. Nilankar Bhanja (Assistant Professor) may be reached at nilankarbhanja2005@gmail.com