Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 4 Issue 4, April - 2017

Design And Implementation Of A Special Purpose Arithmetic Unit For Different Logic Functions

Assistant Prof. Dr. Haitham Kareem Ali
Sulaimani polytechnic University
Technical College of Engineering
Email : haitham_elect@yahoo.com
Sulaimani, Kurdistan Region - Iraq

***Abstract*—The arithmetic and logic unit is the hart of any computer system, but recently it appears that there is a high need for designing new special purposes digital computers directed to the areas of control, communications, and electronics to be used for civic and military applications. In this paper, we designed three virtual logic functions that consist of a number of arithmetic operations (e.g. subtraction, addition, and multiplication). In addition, apart of those functions are control circuits, and in order to keep the design unexpanded, we considered 2-bit numbers. After we completed the required design, we performed a number of tests, and those tests shows the robustness of the proposed design in the sense of using decimal numbers in those logical functions.**

Keywords—Arithmetic unit

1. Introduction

Typically, the ALU has direct input and output access to the processor controller, main memory (random access memory or RAM in a personal computer), and input/output devices. Functionally, an ALU can be divided up into three circuits: the arithmetic circuit, the logic circuit and the shift circuit. The design of the ALU is obviously a critical part of the processor and new approaches to speeding up instruction handling are continually being developed [1,2].

Building a framework for a developing a digital IC design is not easy tasks and consists of a number of stages [3]. This paper presents a novel methodology for building a successful paradigm for designing arithmetic unit, which represents a sharp boundary in abstraction between the physical and logical layers. From the logic level up, the computation consists of a deterministic sequence of zeros and ones. Our approach is applicable for polynomial-base arithmetic circuits design in order to be used as a procedure for graduate students and researcher in related fields to design a special purpose arithmetic unit. Designing a special purpose arithmetic unit is another way of reducing the consumption processing time, since the processing time of built in functions is less than the processing time of the non-built in functions. This will speed up the highly consumption applications (e.g. AutoCad applications).

In this paper, we aim to design three arithmetic units for a different logic functions in the separately form. Those functions have the capability to perform four arithmetic operations and it consists of control circuits to control two different arithmetic operations. The numbers of the functions are assumed to be 2-bit numbers to reduce the size of design. In the next section, we present the proposed design.

2. The proposed design

In this section, we present the three proposed arithmetic logic functions. The three arithmetic units are defined by the following logic functions.

- ArU1= (A^2+2B-D-E)
- ArU2= (AB+2DE-CF)
- Ar.U3= (A*B) ± (2B-C-F)

where, A, B, C, D, and F are 2-bits numbers. We choose different types of functions to show the readers a variety of arithmetic operations. In order to design these arithmetic units, we followed the following procedure. First, we simplified the arithmetic function into terms, and second, we design the logical function for each term. Finally, we combine all the logical functions that correspond to those terms to get the full logical function that corresponds to the arithmetic unit. In the next subsection, we present the first arithmetic unit.

2.1 The design of the arithmetic unit ArU1= (A^2+2B-D-E)

In this subsection, we present the stages for designing the first arithmetic unit. Our design is based on the arithmetic unit according to the first logic function that is given by the following logic equation:

ArU1= (A^2+2B-D-E) (1)

Or

ArU1 = (A*A+B+B-D-E) (2)

Let Z=T-W, where: N=A*A, M=B+B, T=N+M, W=D+E are terms. Let A=A1A0, B=B1B0, D=D1D0, E=E1E0. The logical design of the terms N, M, T, Z

and W are given in Figures (1-6). The steps of design terms shown

(a) N=A*A

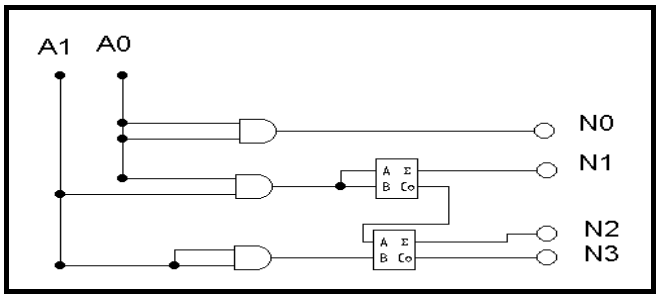

Figure (1) The logic design of the sub-function N

(b) M=B+B

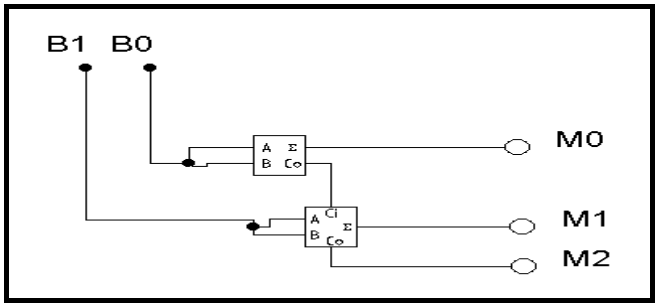

Figure (2) The logic design of the sub-function M

(c) T=N+M

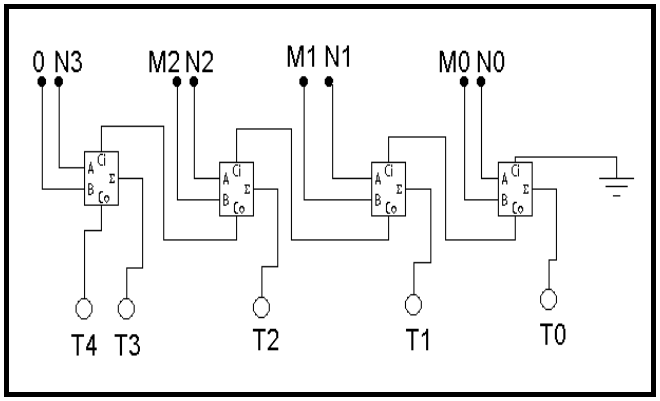

Figure (3) The logic design of the sub-function T

(d) W=E+D

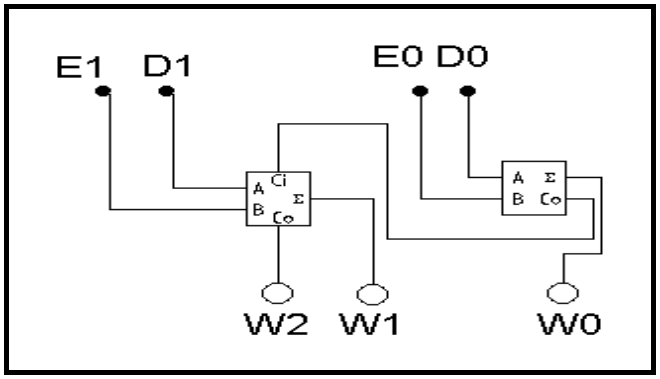

Figure (4) The logic design of the sub-function W

(e) Z=T-W

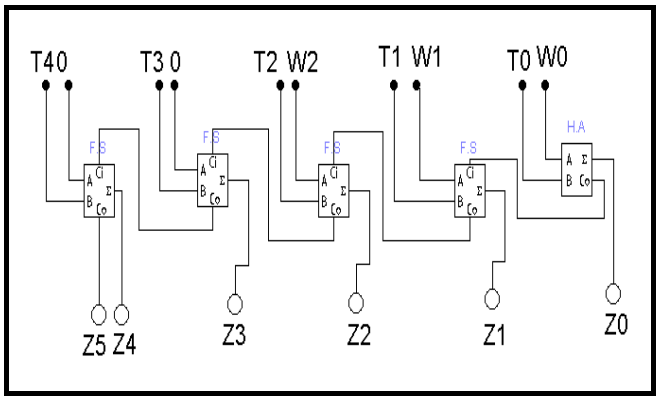

Figure (5) The logic design of the sub-function Z

The following Figure represents the implementation for the design of the first arithmetic unit ArU1.

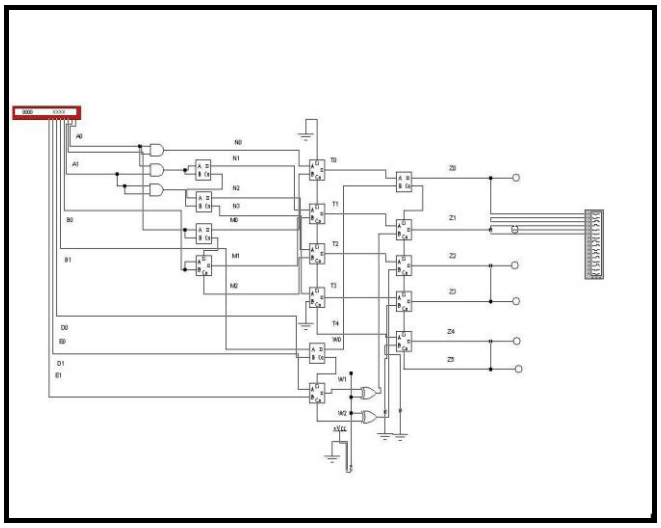

Figure (6) Implementation of first arithmetic unit ArU1

2.2 The design of the arithmetic unit ArU2= (AB+2DE-CF)

In this subsection, we present the stages for designing the 2nd arithmetic unit. Our design is based on the arithmetic unit according to the second logic function that is given by the following logic equation:

ArU2= (AB+2DE-CF) (3)

Or

ArU2= (AB+DE+DE-CF) (4)

Let N=AB, M=DE, W=CF, K=M+M, T=N+K. Then Z=N+2M-W=N+M+M-W and Z=N+K-W = T-W. Let A=A1A0, B=B1B0, C=C1C0, D=D1D0, E=E1E0, F=F1F0. The logical design of the terms K, N, M, T, Z and W are given in Figures (7-13). The steps of design terms are shown below.

(a) N=A*B

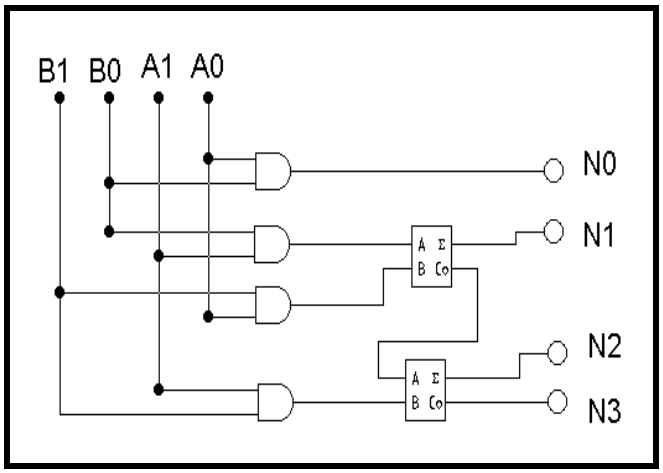

Figure (7) The logic design of the sub-function N

(b) M=DE

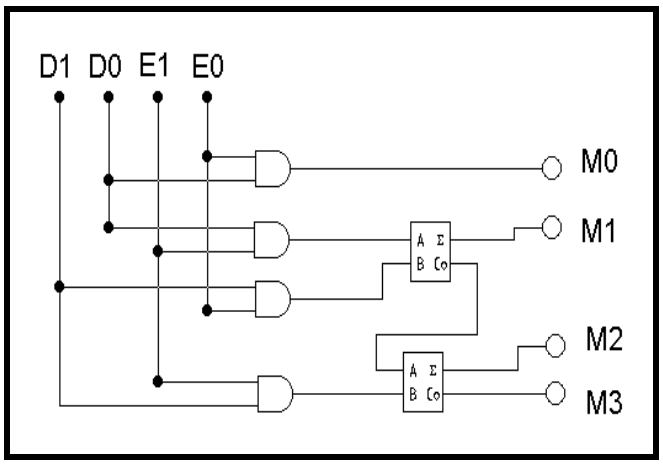

Figure (8) The logic design of the sub-function M

(c) W=CF

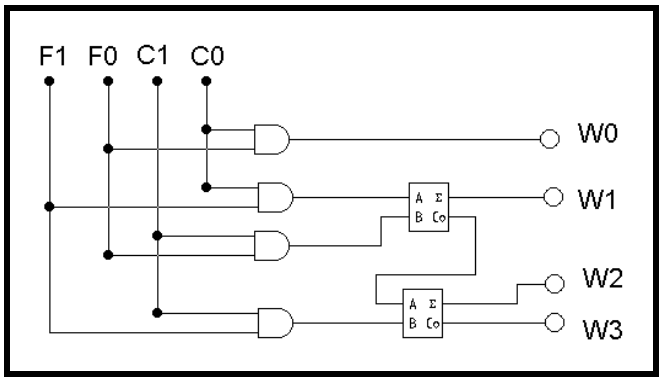

Figure (9) The logic design of the sub-function W

(d) K=M+M

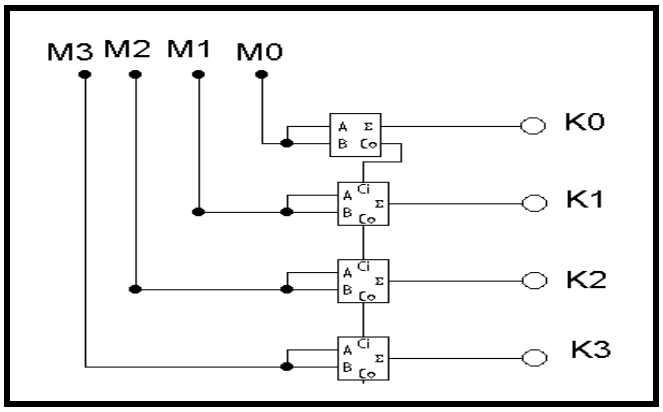

Figure (10) The logic design of the sub-function K

(e) T=N+K

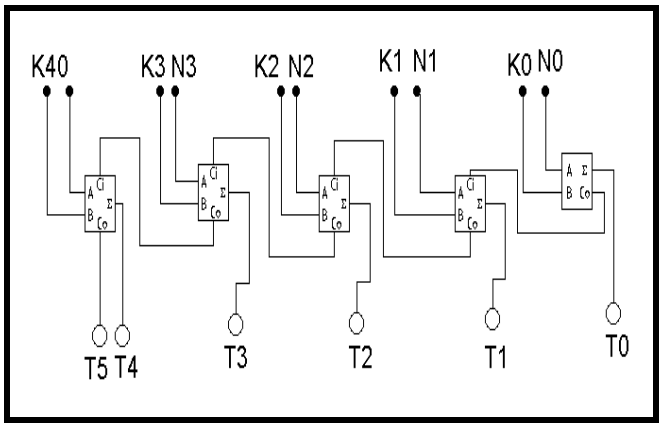

Figure (11) The logic design of the sub-function T

(f) Z=T-W

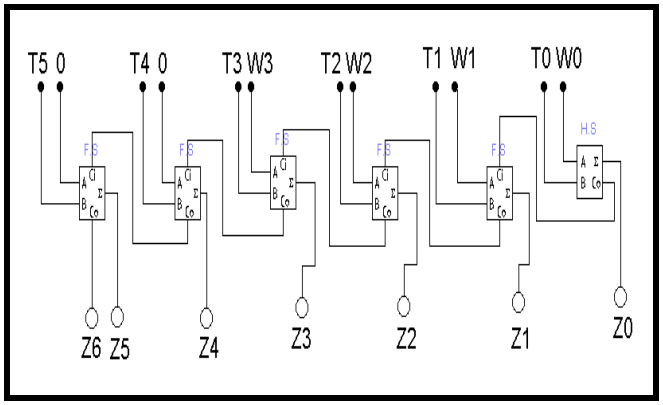

Figure (12) The logic design of the sub-function Z

The following Figure represents the implementation for the design of the second arithmetic unit ArU2.

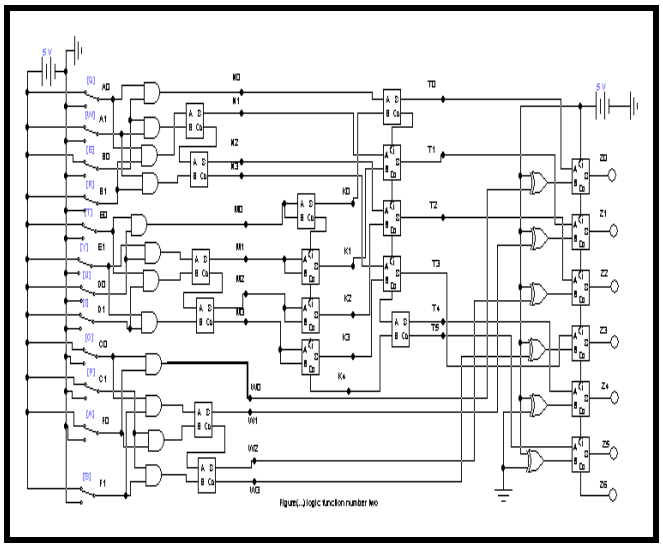

Figure (13) Implementation of second arithmetic unit ArU2

2.3 The design of the arithmetic unit ArU3= (A*B)±(2B-C-F)

In this subsection, we present the stages for designing the 3rd arithmetic unit. Our design is based on the arithmetic unit according to the first logic function that is given by the following logic equation:

ArU3= (A*B)±(2B-C-F) (5)

Z = [(M) ± (B+B-(C+F))]

Z = [M ± (D-N)]

Z = [M±K]

where: M=A*B, N=C+F, D=B+B, K=D-N, Z=M±K. Let A=A1A0, B=B1B0, C=C1C0, F=F1F0, where the logical design of the terms K, N, M, D, and Z are given in Figures (14-19), where the sign (±) means that there is a control line between the addition and subtraction operation. The steps of the design terms are shown below.

(a) M=A*B

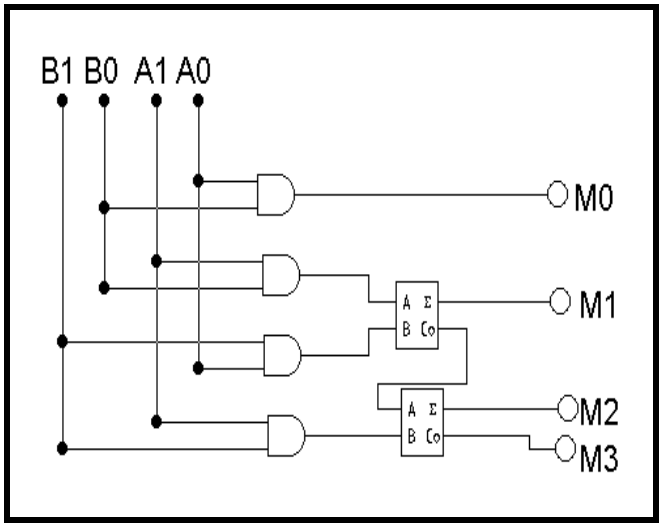

Figure (14) The logic design of the sub-function M

(b) N=C+F

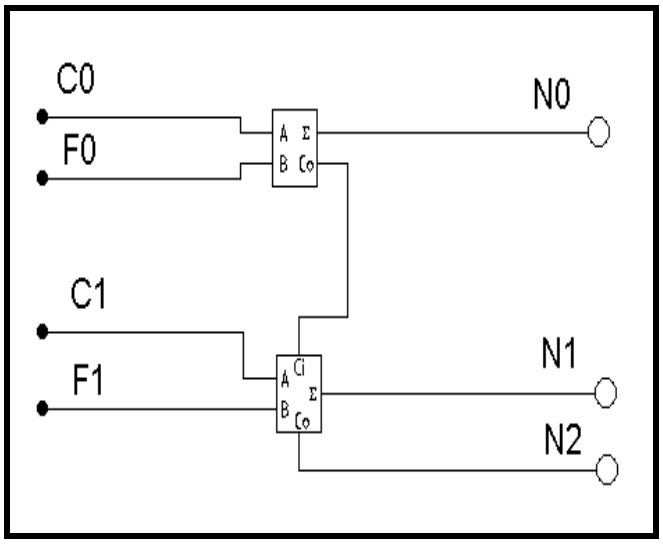

Figure (15) The logic design of the sub-function N

(c) D=B+B

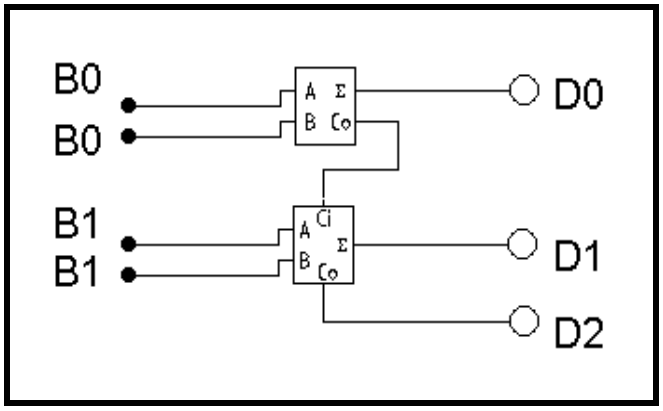

Figure (16) The logic design of the sub-function D

(d) K=D-N (S=1 subtraction)

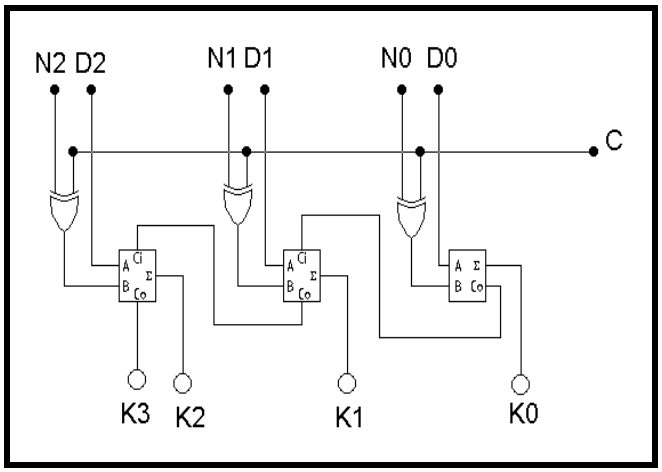

Figure (17) The logic design of the sub-function K

(e) Z=M±K (x=0 (addition) and x=1 (subtraction)

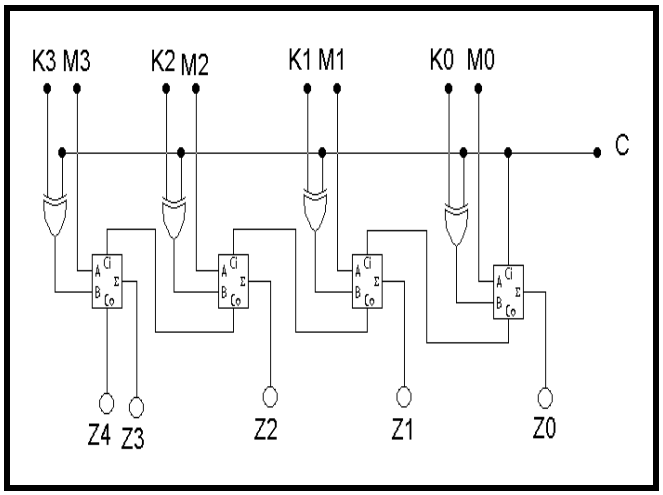

Figure (18) The logic design of the sub-function Z

The following Figure represents the implementation for the design of the third arithmetic unit ArU3.

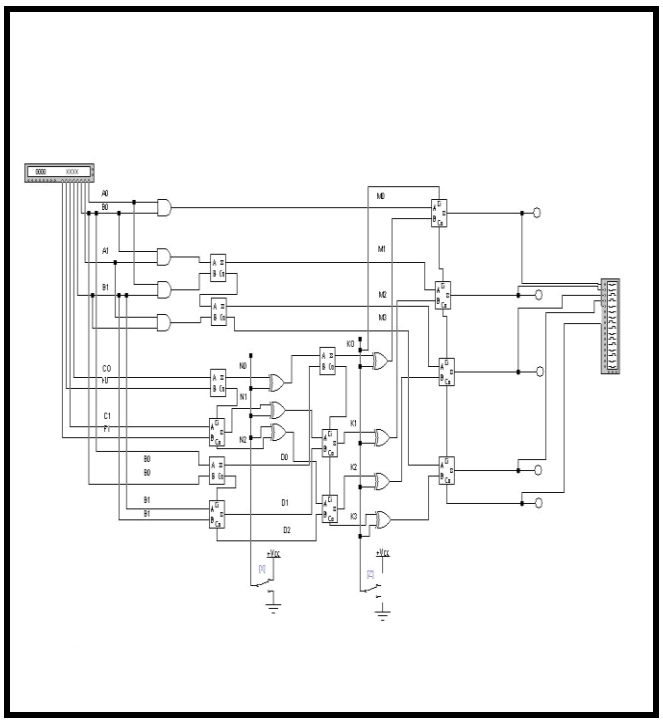

Figure (19) Implementation of third arithmetic unit ArU3

3. The testing and results

In this section, we present the testing of the proposed arithmetic units after implementations. We considered two input cases for each number, and the output (results) are shown in the following tables.

(a) The results of implementing ArU1 shown in the table (1).

Table (1) result of the ArU1

variable	A	B	D	E
Input Case 1	11	11	11	11
Result1	001001			
Input Case 2	10	10	10	01
Result2	000101			

(b) The results of implementing ArU2 shown in the table (2).

Table (2) result of the ArU2

variable	A	B	C	D	E	F
Input Case1	11	11	11	11	11	11
Result1	0010010					
Input Case2	10	10	10	01	01	01
Result2	0000100					

(c) The results of implementing ArU3 shown in the table (3).

Table (3) result of the ArU3

variable	A	B	C	F
Input Case1	11	10	01	01
Result1	01000 if (+) or 00100 if (-)			
Input Case2	10	10	00	00
Result2	01000 if (+) or 00000 if (-)			

5. Conclusions

In this paper, we designed three virtual logic functions that consist of a number of arithmetic operations (e.g. subtraction, addition, and multiplication). In addition, a part of those functions are control circuits, and in order to keep the design unexpanded, we considered 2- bit numbers. After we completed the required design, we performed a number of tests, and those tests shows the robustness of the proposed design in the sense of using decimal numbers in those logical functions.

The References:

[1] Thomas L. Floyd, ' Digital Fundamentals ', Ninth Edition, Prentice Hall, Pearson Education International (2006).

[2] M. Morris Mano and Charles R. Kime, 'Logic and Computer Design Fundamentals', Prentice Hall, Inc., 2008.

[3] Dhafer R. Zaghar, 'Design and Implementation of IF Digital Receiver', PH.D. thesis, Electrical Engineering-Baghdad University, 2002.