UACEE International Journal of Advances in Electonics Engineering Volume 2 : Issue 1 ISSN 2278 - 215X (Online)

Hardware Efficient Reconfigurable Arithmetic Unit

Prithvi Chawla
Electronics Engineering Department
Yashwantrao Chavan College of Engineering
Nagpur, Maharashtra
prithvichawla@gmail.com

Prasanna Palsodkar
Electronics Engineering Department
Yashwantrao Chavan College of Engineering
Nagpur, Maharashtra
palsodkar.prasanna@gmail.com

Shabul Sheikh
Electronics Engineering Department
Yashwantrao Chavan College of Engineering
Nagpur, Maharashtra
shabul.sheikh@gmail.com

***Abstract*—In this paper we present an arithmetic unit which performs addition and subtraction on Binary & Binary Coded Decimal(BCD) numbers. The unit is able to perform effective addition-subtraction operations on unsigned, signed magnitude and various complement representations. The design is runtime reconfigurable and all the subunits have been designed to work with least delay. The proposed unit is synthesized for 4vfx60ff672-12 Xilinx Virtex-4 FPGA.**

Keywords- Binary / BCD operations, FPGA, universal arithmetic unit.

I. INTRODUCTION

Various applications, e.g. commercial and financial electronic transactions, internet and industrial control require precise arithmetic for different data representation formats. Such application cannot tolerate errors of conversion between binary and decimal formats. If a binary approximation is used instead of an exact decimal fraction, result can be incorrect even if subsequent arithmetic is correct. When performing decimal operation on traditional binary based hardware, Excessive delays are introduced. Therefore, decimal arithmetic is necessary in many financial and commercial applications that process decimal values and perform decimal rounding. This paper deals with the design of a single architecture that can perform arithmetic operations on binary and BCD notation. It also supports both signed and unsigned operations for binary and BCD representation. The proposed structure uses the effective addition/subtraction approach for binary, BCD and single precision notations.

The remainder of this paper is organized as follows. Section 2 outlines the background for BCD arithmetic and presents the related work for Binary and BCD Arithmetic. Section 3, exhibits the details of Reconfigurable Universal Adder design.

II. METHODOLOGY

A. BCD arithmetic

8421 BCD is a way to express each of the decimal digits with binary code. There are only first ten 4-bit binary code groups (0000_2 to 1001_2) are used to express each decimal digit from 0 to 9. The designation 8421 BCD code indicates binary weights of the four bit (2^3, 2^2, 2^1, 2^0). The remaining 4-bit binary code groups (1010_2 to 1111_2) for decimal digit from 10 to 15 are left unused when decimal computing is considered. Assuming two decimal numbers are added using a 4-bit binary adder. The following steps are performing for BCD addition:

1. Add two 4-bit BCD numbers (equivalent to decimal digit) using binary addition.

2. If 4-bit sum is equal to or less then 9, the sum is valid BCD number and no correction is needed.

3. If the 4-bit sum is greater than or if a carry is generated from the sum is invalid BCD number. Then, the digit 6 (0110_2) should be added to the sum to correct the invalid BCD representation.

In case of decimal subtraction, additional processing needed for 10’s complement of the subtrahend. As the BCD code do not include a code for decimal digit 10 and for this reason a nine’s complement representation is used and generate carry from the sum is again added to the sum.

B. Related Work for Binary/BCD

There is a wide range of literature available in field of BCD arithmetic. Some of the first contributions were made by Schmookler et. al.[1] and Adiletta et. al. [2]. An approach towards to architecture dealing with both BCD and binary was shown by Levine et. al.[3] and Anderson[4], while one of the first BCD sign-magnitude adder/subtractor architecture was presented by Grupe [5]. An area efficient sign magnitude adder was later developed by Hwang [6]. Area occupied by this design was least amongst all the previous designs.In this approach two additional conversions are introduced before and after the binary novelty in Hwang's proposal comes with the separation of the binary and the decimal results, using a multiplexer to select the correct output as is depicted in Figure 1.

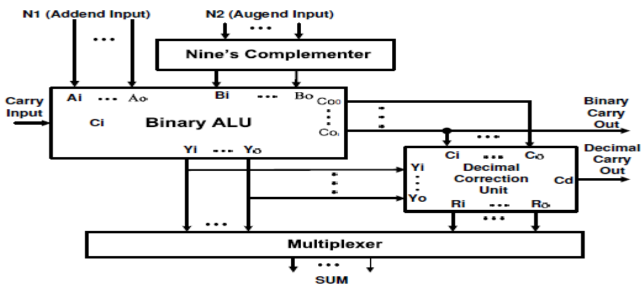


Figure 1. Hwang's proposal

Fischer et al. [7] (Fig. 2) later came up with a compact design that employed only one adder but the latency was a problem as it had to use an additional correction block.This unit encodes the incoming operands as well as the adder result to achieve the desired functionality.

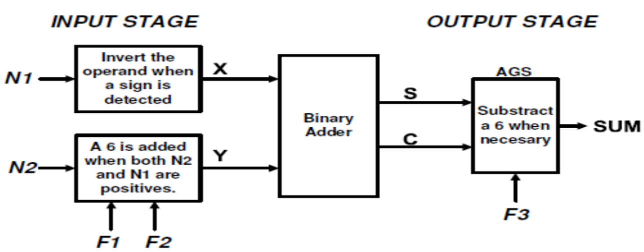


Figure 2. Fischer's proposal

The approach to construct BCD architectures in many IBM processors is based on the work presented by Haller et. al. in [8]. This architecture shown in Fig. 3 operated in a single cycle, though requiring corrections in some cases. In the case of subtraction there is a need for the computation of the complement after the subtraction to obtain the correct difference, hence increasing the latency. Another improvement in the same architecture was the optimization of the carry chain resulting in a slight delay improvement with an increased area of the unit.

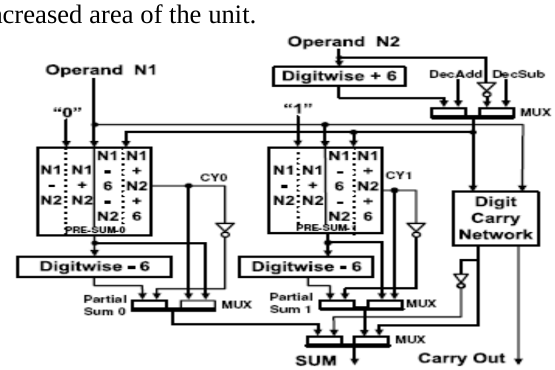


Figure 3. Haller's proposal

The design of the Universal Adder (Fig. 4) proposed by D.R.Humberto et al. [9] uses effective addition/subtraction operations on unsigned/sign-magnitude, and various complement representations. This design overcomes the limitations of previously reported approaches that produce some of the results in complement representation when operating on sign-magnitude numbers. This design proposed that the major disadvantage of the previous designs i.e. having the subtrahend the smaller number in magnitude, was eliminated by their approach. This paper proposed a new method without the essential use of Inputs Outputs complement to perform BCD subtraction.

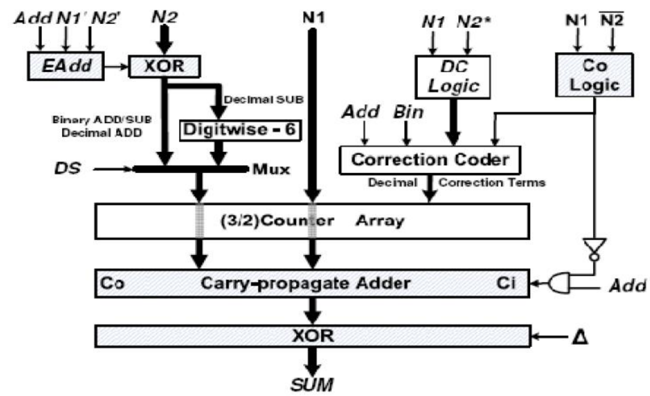


Figure 4. Humberto's Proposal

Sreehari et al. recently came up with the prefix logic based BCD adders and proposed a novel unified BCD binary adder-subtractor [10] which is considered as the fastest unified adder in the literature so far. The architecture is divided into three major parts, the pre-computation stage, prefix network and post computation stage illustrated in Fig.5. The pre-computation block consists of logic to compute propagate and generate signals for both BCD and Binary addition/subtraction with much lower latency.

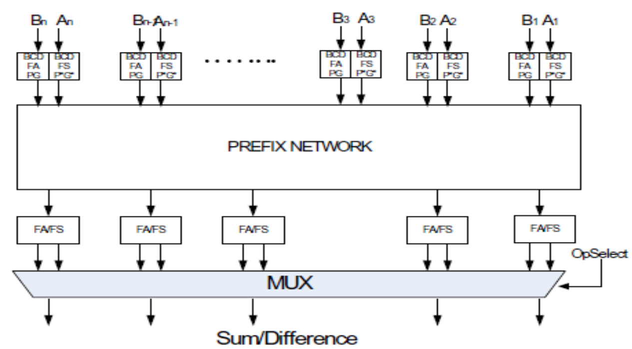


Figure 5. Sreehari's Proposal

III. PROPOSED REDUCED DELAY UNIVERSAL ADDER

The whole architecture of the reduced delay universal adder sub- divided into six subunits. Sixth subunit is Carry in circuitry for both carry-propagate-adder including Co logic works parallel to the rest design.

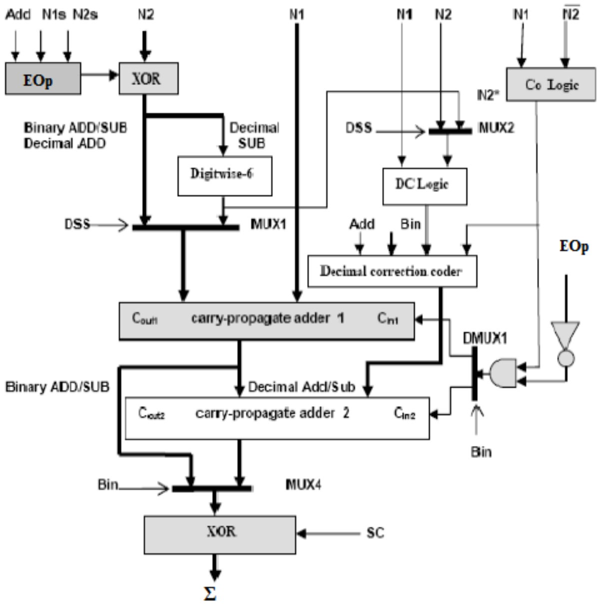


Figure 6. Proposed Architecture

The first subunit of our proposal includes Eadd logic, XOR and Digitwise-6 logic while second subunit includes DC logic and decimal correction coder. Third and fourth subunits are carry-propagate-adder 1 and carry-propagate-adder 2 respectively. The SUM correction SC logic divides as a fifth subunit.

In order to perform binary arithmetic, only first, third, and fifth subunits works together in pipelined manner while sixth subunit works parallel to the rest units. Digitwise-6 logic, second and fourth subunits are left unused. And in order to perform decimal operation, first five subunit works together in pipelined manner while sixth subunit in parallel. Second subunit follows first subunit and fourth subunit follows third subunit.

The main objective of our design was to reduced delay of binary and BCD adder/subtrator operations using parallelizing and pipelining scheme. To achieve this we used the Humberto [9] design as a base. We kept its functionality as it is, with minimizing its delay. For unsigned and sign-magnitude addition/subtraction or effective addition/subtraction, we also use s/370 sign-magnitude adder as a base.

A. BINARY ARITHMETIC

Let us assume N1 and N2 being two n-bit sign magnitude numbers, such that $N1 = [N1_{n-1}\ N1_{n-2}...N1_0]$ and $N2 = [N2_{n-1}N2_{n-2}...N2_0]$, with $N1_{n-1}$ and $N2_{n-1}$ used as sign bits of binary or BCD representations.

Consider the operation R = N1 Op N2 (1)

with input signal Op indicates desirable addition/ subtraction operation and R being the result of the operation.

Op = 0 for addition &

Op = 1 for subtraction

Conclusion is that the addition/subtraction of two unsigned/signed- magnitude numbers can be performed with the determination of effective operation.

The effective operation can be computed by the following Boolean expression:

$$EOp = (N1_{n-1} \oplus N2_{n-2} \oplus \overline{Op}) \quad (2)$$

If EOp is equal to one, effective addition will be performed, and if EOp is equal to zero, effective subtraction will be performed. The effective operations are described in Table.1

N1s	N2s	Op	EOp	Operations
0	0	0	1	+(|N1|+|N2|)
0	0	1	0	+(|N1|-|N2|)
0	1	0	0	+(|N1|-|N2|)
0	1	1	1	+(|N1|+|N2|)
1	0	0	0	-(|N1|-|N2|)
1	0	1	1	-(|N1|+|N2|)
1	1	0	1	-(|N1|+|N2|)
1	1	1	0	-(|N1|-|N2|)

Table 1

The effective operation i.e.addition/subtraction operation is performed on the absolute values of N1 and N2, denoted by |N1| and |N2| respectively, and the operation produces no overflow.

Assuming that the operation is binary effective addition then the following equation establishes binary effective addition:

$$SUM= |N1|+|N2|+ Co \quad (4)$$

Where, Co is used to find whether |N1| is greater than |N2| or less than/equals to |N2|. Co should be equals to zero for effective addition. Boolean expression for Co can be defined as following:

$$Co= G_{0n-1} [|N1|, \overline{|N2|}]$$

$$=G_{n-1}| (P_{n-1}.G_{n-2})|...| (P_{n-1}.P_{n-2}...P_{n-i}.G_i)|.....|(P_{n-1}.P_{n-2}......P_1.G_0)$$

Where $G_i = N1_i.\overline{N2_i}$ and $P_i=N1_i|\overline{N2_i}$ are generate and propagate signals respectively.

Assume that the operation is binary effective addition.Then the following equation establishes the equation

$$\Sigma = |N1|+|N2| \quad (3)$$

Assume that the operation is binary effective subtraction. Then the equation is represented as

$$\Sigma= |N1|+\overline{|N2|}+ Co \quad (4)$$

Where Co is used to find whether |N1| is greater than |N2| or less than/equals to |N2|.

As it is stated that there is no overflow, so Co is given as

$$Co = G_0^{n-2}\,[N1,\overline{N2}] \quad (5)$$

Where G_0^{n-2} indicates the group generate signal from bit 0 to bit n-2.Co can be expanded as

$$Co = G_{n-1}|\,(P_{n-1}.G_{n-2})|...|\,(P_{n-1}.P_{n-2}\ldots P_{n-i}.G_i)|\ldots$$
$$|\,(P_{n-1}.P_{n-2}\ldots\ldots P_1.G_0) \quad (6)$$

Where $G_i = N1_i.\overline{N2}_i$ and $P_i=N1_i|\overline{N2}_i$ are generate and propagate signals respectively.

For binary effective subtraction following possibilities can be defined:

Case 1: If |N1|>|N2| then Σ will be positive and the following equation establishes binary effective subtraction:

$$\Sigma = |N1|+|N2|+ Co \quad (7)$$

Where Co =1 because |N1|>|N2| and performs 2's complement subtraction.

Case 2: if |N1|<|N2| then Σ will be negative and the following equations establish binary effective subtraction:

$$\Sigma = |N1|+|N2|+ Co \quad (8)$$

Where Co=0 because |N1|<|N2| and Σ itself in one's complement representation to represent negative Σ.

Case 3: if |N1|=|N2| then sum will be zero and the following equations establish binary effective subtraction:

$$\Sigma = |N1|+|N2|+ Co \quad (9)$$

Where Co=0 because |N1|=|N2|.

In order to generate a correct sign-magnitude result, an additional correction step ⊕SC is used. The final magnitude result becomes:

$$|\Sigma| = \Sigma \sum_{k=0}^{n-2} \oplus SC \quad (10)$$

The SC is computed as follows:

$$SC = \overline{Co}.\overline{EOp} \quad (11)$$

Finally, the sign bit of the result is updated as shown equation

$$\Sigma = [N1_{n-1} \oplus SC] \quad (12)$$

B. BCD ARITHMETIC

In this section we describe in more details of additional additions to the original binary adder needed for decimal addition/subtraction operations.

Assume that the operation is decimal effective addition. Then the following equation establishes decimal effective addition:

$$\Sigma = |N1|+|N2|+0110_2 + 0 \text{ (if DC=1)} \quad (13)$$

$$\Sigma = |N1|+|N2| + 0 + 0 \text{ (if DC=0)} \quad (14)$$

Assume that the operation is decimal effective subtraction and all possibilities can be defined as following:

Case 1: If |N1|>|N2| then SUM will be positive and the following equation establishes decimal effective subtraction:

$$\Sigma = |N1|+|N2^*|+ 0110_2+1 \text{ (if DC=1)} \quad (15)$$

$$\Sigma = |N1|+|N2^*| + 0 + 0 \text{ (if DC=0)} \quad (16)$$

Where '*' indicates nine's complement of the operand.

Case 2: if |N1|<|N2| then SUM may be negative and the following equations establish decimal effective subtraction:

$$\Sigma = |N1|+|N2^*|+ 1100_2 + 0 \text{ (if DC=1)} \quad (17)$$

$$\Sigma = |N1|+|N2^*|+0110_2 + 0 \text{ (if DC=0)} \quad (18)$$

Case 3: if |N1|=|N2| then sum and DC signal will be zero and the following equations establish decimal effective subtraction:

$$\Sigma = |N1|+|N2^*|+0110_2 + 0 \quad (19)$$

The digital carry logic (DC) signal for decimal operations are obtained as follows:

$$DC=A|B.Cin \quad (20)$$

Where

A=G3|P3.P2|P3.P1|G2.P1|P3.G0|G2.G0

B=P3|G3|P2.G1

In the proposed adder illustrated in Figure 6,EOp logic controls one's complement operation of the subtrahend in binary and decimal operations. The nine's complement computation for the subtrahend is performed using one's complement then "DigitWise-6" (*DW*) hardwired logic, as used in many previous designs. The DW value ND = N2−6_{10} is obtained with the following equations that modify each bit of the BCD nibble as follows:

$ND[3] = \overline{N2[3]} \cdot \overline{N2[2]}|\overline{N2[3]} \cdot \overline{N2[1]}|N2[3] \cdot N2[2] \cdot N2[1]$

$ND[2] = N2[2] \oplus N2[1]$

$ND[1] = \overline{N2[1]}$

$ND[0] = N2[0]$

Please note that when a binary operation is performed Bin = 1 the decimal correction term is not needed and left inactive, otherwise decimal operations are performed. The reduced delay binary and BCD adder is set up with the aforementioned logic, two carry-propagate-adder, some multiplexers and de-multiplexers, and a set of XOR gates. The final organization is depicted in Figure 6. Note that the input N2* for computing the digit carry logic is equal to nine's complement when processing decimal subtraction. When processing decimal addition N2* is equal to N2 otherwise left inactive. The multiplexer signal control for decimal subtraction or any addition (*DSS*) is computed by:

$$DSS = \overline{EOp} \cdot \overline{Bin} \quad (21)$$

The final complement operation is controlled by equation

$$SC = (\overline{Co} \cdot \overline{EOp})$$

IV. EXPERIMENTAL RESULTS ANALYSIS

The sub units of proposed Universal Adder was implemented using Verilog HDL, synthesized, functionally tested, and evaluated using the ISE 12.2 Xilinx design tools targeting 4vfx60ff672-12 VIRTEX 4 FPGA device.

Figure 7. RTL View of DSS Logic Block and its Waveform

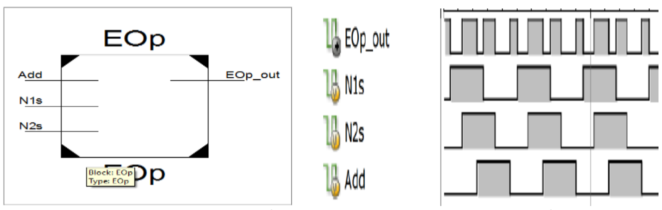


Figure 8. RTL View of EOp Logic Block and its Waveform

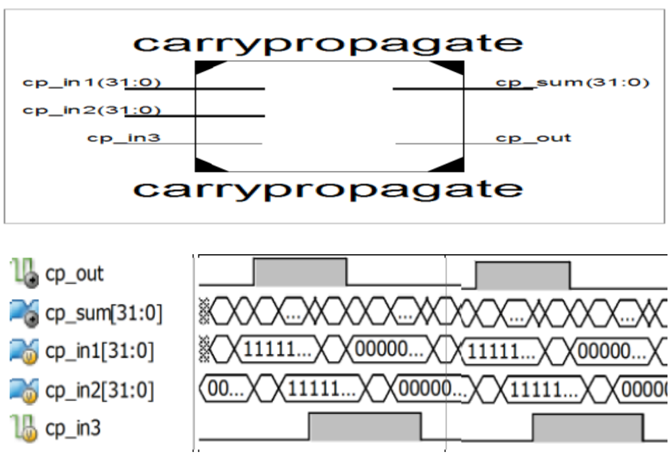
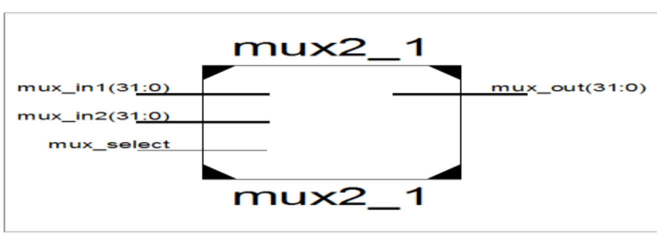


Figure 9. RTL view of Carrypropagate Adder & its Waveform



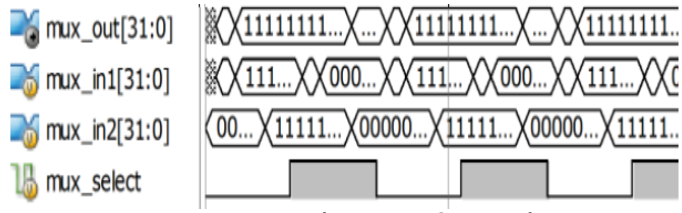


Figure 10. RTL View of 2:1MUX and its Waveform

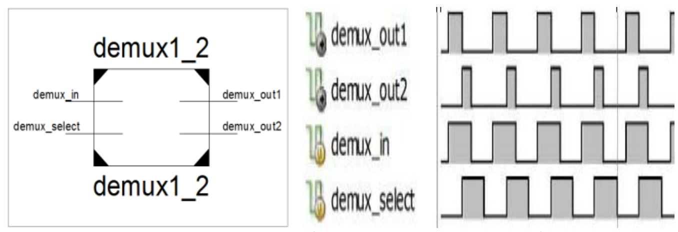


Figure 11. RTL View of 1:2 DeMux and its Waveform

REFERENCES

[1] M.S.Schmookler and A. Weinderger. "Decimal Adder for Directly Implementing BCD Addition Utilizing Logic Circuitry", *International Business Machines Corporation,US patent 3629565*, pages 1 – 19, Dec 1971.

[2] M. J. Adiletta and V. C. Lamere. "BCD Adder Circuit". *Digital Equipment Corporation, US patent 4805131*, pages 1 – 18, Jul 1989.

[3] S. R. Levine, S. Singh, and A. Weinberger. Integrated Binary-BCD Look-Ahead Adder. *International Business* Machines Corporation, US patent 4118786, pages 1–13, Oct 1978.

[4] J. L. Anderson. Binary or BCD Adder with Precorrected Result. Motorola, Inc., US patent 4172288, pages 1–8, Oct 1979.

[5] U. Grupe."Decimal Adder", Vereinigte Flugtechnische Werke-Fokker gmbH, US patent 3935438, pages 1 – 11, Jan 1976.

[6] S. Hwang. "High-Speed Binary and Decimal Arithmetic Logic Unit", *American Telephone and Telegraph Company, AT&T Bell* Laboratories, US patent 4866656, pages 1-11, Sep 1989.

[7] H.Fischer andW. Rohsaint. "Circuit Arrangement for Adding or Subtracting Operands Coded in BCD-Code or Binary-Code", *Siemens Aktiengesellschaft, US patent 5146423*, pages 1 – 9, Sep 1992.

[8] W. Haller, W. H. Li, M. R. Kelly, and H. Wetter. "Highly Parallel Structure for Fast Cycle Binary and Decimal Adder Unit". International Business Machines Corporation, US patent 2006/0031289, pages 1 – 8, Feb 2006.

[9] D.R.Humberto Calderón, G. N. Gaydadjiev, S. Vassiliadis, "Reconfigurable Universal Adder", Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP 07), pages 186-191, July 2007.

[10] Sreehari Veeramachaneni, M, Kirthi Krishna; V, Prateek G, S. Subroto, S, Bharat, M.B.Srinivas, "A Novel Carry-Look Ahead Approach to a Unified BCD and Binary Adder/Subtractor", 21st International Conference on VLSI Design 2008, pages 547-552, Jan2008.

[11] Mohit Tyagi, Apurva Vaishist, Kavita Khare, "A Novel Hardware Efficient Reconfigurable 32-Bit Arithmetic Unit for Binary, BCD and Floating Point Operands" ISSN : 0975-5462 Vol. 3 No. 5 May 2011.