INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

Carry-Look ahead Approach to a Unified BCD and Binary Adder/Subtractor

Rajesh V[1], Kavitha V[2], Gowtham M[3], Bazariya Begum I[4]

[1] PG Scholar, [2] PG Scholar,[3] PG Scholar,[4] PG Scholar,
[1234]SNS College of Technology,
Coimbatore-641035,
[1]vr.rajesh91@gmail.com,[2]kavitha0728@gmail.com,[3]1555gowthamm@gmail.com,
[4]bazariyabegum27490@gmail.com

Abstract: *Increasing prominence of commercial, financial and internet-based applications, which process decimal data, there is an increasing interest in providing hardware support for such data. In this paper, new architecture for efficient binary and Binary Coded Decimal (BCD) adder/sub tractor is presented. This employs a new method of subtraction unlike the existing designs which use 10's complements, to obtain a much lower latency. Though there is a necessity of correction in some cases, the delay overhead is minimal. A complete discussion about such cases and the required logic to process is presented. The architecture is run-time reconfigurable to facilitate both BCD and binary operations, including signed and unsigned numbers. The proposed circuits are compared (both qualitatively as well as quantitatively) with the existing circuits in literature and are shown to perform better. Simulation results show that the proposed architecture is at least 11% faster than the existing designs.*

Keywords: *Binary coded decimal, latency, run-time.*

1. INTRODUCTION

Due to growing importance of decimal arithmetic in commercial, financial and internet-based applications, which cannot tolerate errors from converting between binary and decimal formats, hardware support for decimal arithmetic is receiving an increased attention. Despite the widespread use of binary arithmetic, decimal computation remains essential for many applications. Not only it is required whenever numbers are presented for human inspection, but it is also often a necessity when fractions are involved.

Decimal fractions are pervasive in human endeavours, yet most cannot be represented by binary fractions. The value 0.1, for example, requires an infinitely recurring binary number. If a binary approximation is used instead of an exact decimal fraction, results can be incorrect even if subsequent arithmetic is correct. The major consideration while implementing Binary Coded Decimal (BCD) arithmetic will be to enhance its speed as much as possible which is being addressed in this paper. But to facilitate even binary applications on the same hardware a reconfigurable approach needs to be adopted.

This paper deals with the design of an architecture that can perform both binary and BCD addition/subtraction. It also supports both signed and unsigned operations. All the existing architectures use 10's or 9's complement to implement subtraction in BCD. But this has been found to have a very high latency hence a new approach has been proposed to overcome this problem. The architecture has been designed to have maximum hardware utilization. The proposed algorithm for the unified BCD and binary adder/subtracor is given. In the proposed architecture is presented. Simulation results for the proposed and existing circuits and comparisons are carried out. Finally a conclusion is presented.

2. EXISTING METHOD

A Hwang's Proposal

An area efficient sign magnitude adder was later developed by Hwang. In his approach two additional conversions were introduced before and after the binary addition. Area occupied by this design was least amongst all the previous designs.

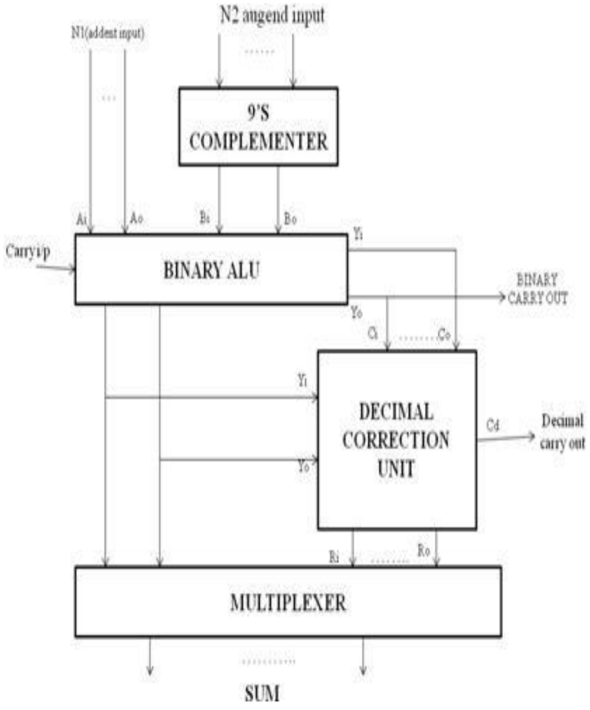

Figure 1: Hwang's proposal

B Fischer's Proposal

The use of duplicate hardware to compute the output in the presence of a carry and in its absence. It then selects the appropriate one as the carry and result is computed was improvised by Fischer where only a single adder was employed to reduce the area over-head. But there was a higher latency due to the additional correction block employed. 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.

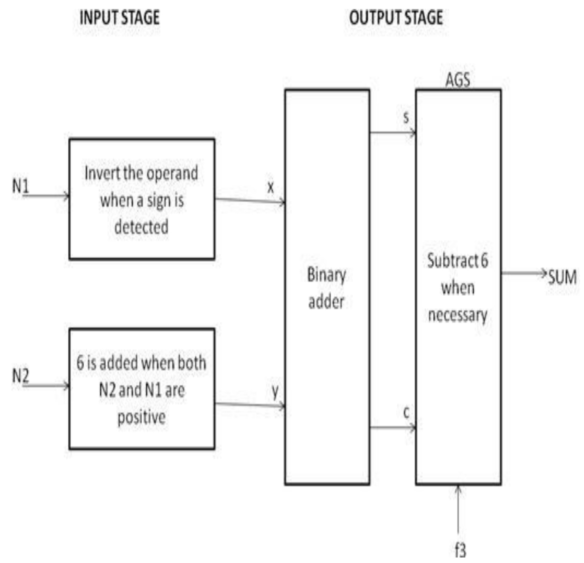
INPUT STAGE
OUTPUT STAGE
AGS
N1
Invert the operand when a sign is detected
x
N2
6 is added when both N2 and N1 are positive
y
Binary adder
s
c
Subtract 6 when necessary
SUM
f3

Figure 2: Fischer's Proposal

C Humberto's Proposal

The Universal adder micro architecture design proposed by Humberto uses effective addition/subtraction operations on unsigned, sign-magnitude, and various complement representations. This design also overcomes the limitations 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.

One of the major points to note is that all these proposals make use of complement addition to perform subtraction

3. PROPOSED ALGORITHM

The proposed algorithm aims at performing both BCD and binary addition/subtraction. The major concern is to avoid 10's complement to perform subtraction which is the reason for the high latency in the existing architectures. The proposed design can be divided into three major parts, the pre-computation stage, the prefix network and the post computation stage.

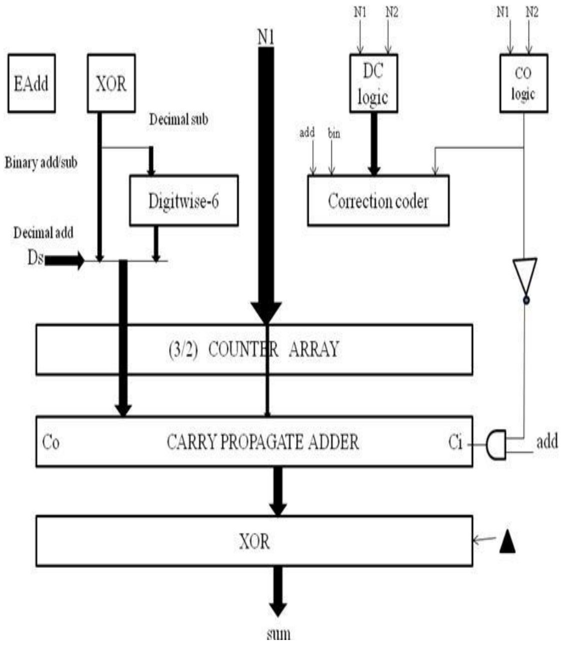

Figure 3: Humberto's Proposal

A Pre-Computation Stage

The pre-computation stage generates control signals named propagate (P) and generate (G). These control the operation of the prefix network. These control signals are generated for every significant stage in the N-digit number i.e. there 2*N(P*N + G*N) control signals. These denote whether the kth stage propagates the carry/borrow signal or generates it respectively In the case of addition of BCD digits A and B (two digits at the kth stage) the following equations denote propagate and generate:

$$P \Rightarrow A+B=9$$
$$G \Rightarrow A+B>9$$

B Prefix Network

These propagate and generate bits are sent to the prefix network which has a network of blocks which calculates the group propagate and generate bits. The group Pk: 0 and G k:0 bits denote whether the first k stages propagate or generate the carry/borrow.

C Post-Computation Stage

These bits are sent to the post-computation blocks which compute the final sum/difference based on these bits. In case of binary, after obtaining the carry/borrow the sum/difference is the XOR of the carry/borrow and the propagate bit.

In case of BCD, it is a little complicated. As mentioned, for the addition of two BCD digits if there is an overflow then a correction value of 0110(6) has to be added. For the subtraction of BCD digits, all the existing architectures are employing 10's complement subtraction. But computing 10's complement induces a very high latency in the operation.

Hence the proposed architecture uses propagate and generate bits defined specifically for subtraction to compute the borrow bits for every stage using the prefix network. After the borrow bit is computed, the individual digits at every significant stage are subtracted by using 2's complement using 4-bit binary adders. But then if a borrow output is generated then the output has to be corrected by adding 1010 (10) to the difference. This is checked by inspecting the borrow input of the subsequent stage which has already been generated by the prefix network.

4. ARCHITECTURE OF THE PROPOSED BCD AND BINARY ADDER/SUBTRACTOR

The proposed architecture performs both BCD and binary addition/subtraction including signed and unsigned numbers. This architecture can be divided into three major parts, the pre-computation stage, the prefix network and the post-computation stage. This architecture is illustrated by a block diagram.

For the case of BCD computation it can be observed from the diagram that the pre-computation block for every significant stage consists of logic to generate (P, G) and (P*, G*). Depending on whether addition or subtraction is selected the corresponding propagate and generate bits are sent to the prefix network using an array of multiplexers.

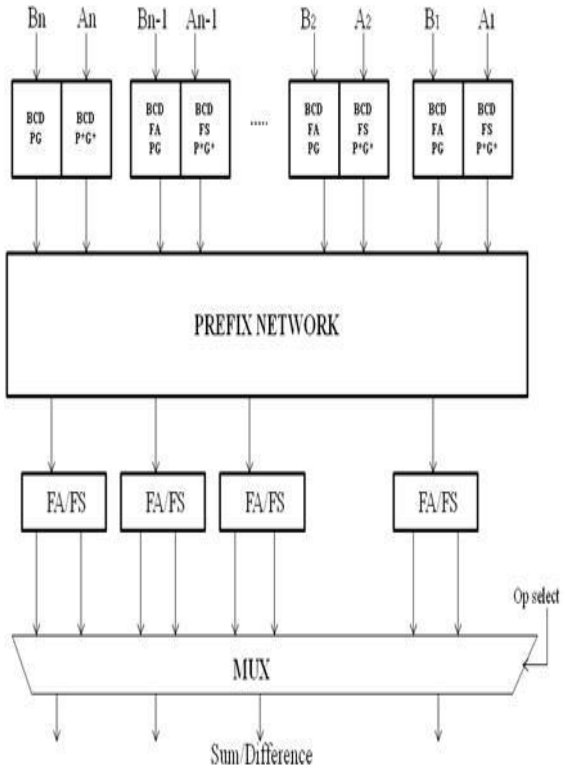

Figure 4: Block Diagram of the proposed architecture

The selection of the prefix network can be made according to the requirements of area, power and delay from the wide range available in literature. This generates the group propagate and generate which when combined with the carry/borrow input generates carry/borrow for every stage. These bits are taken by the final post-computation BCD Full Adder/Subtractor.

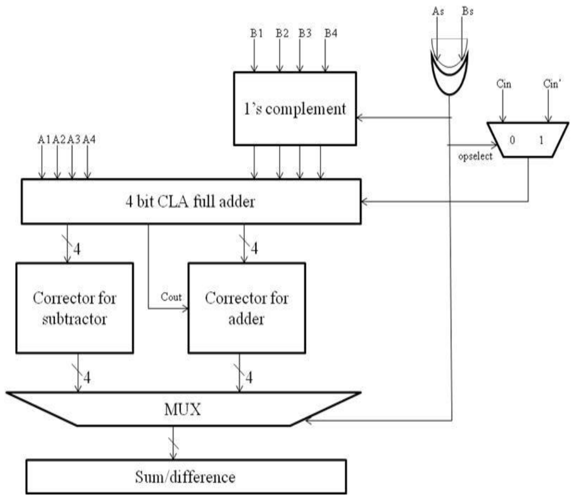

Figure 5: Block Diagram for BCD Adder/Subtractor

The BCD Full Adder/Subtractor computes the sum if selected by the control signal. The addition operation is performed by adding the two BCD digits using the 4-bit binary carry look-ahead adder and the correction block. This diagram is shown in Figure.

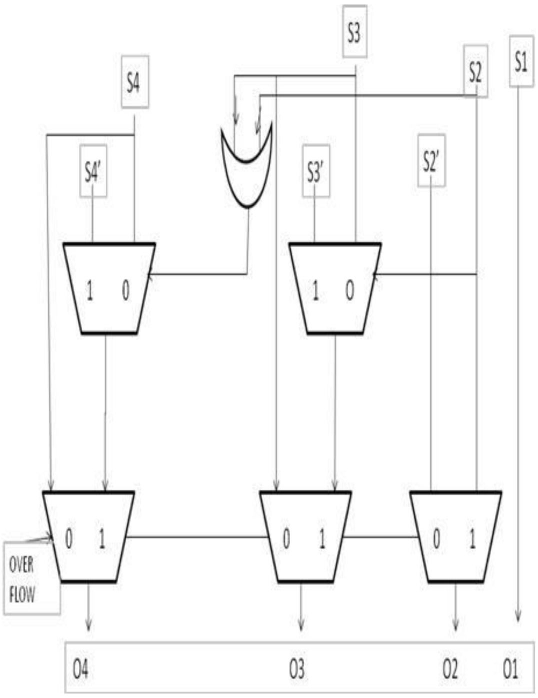

Figure 6: Correction Block for BCD Adder

The subtraction operation is done by 2’s complement addition. But the difference generated needs to be corrected. The borrow input for the subsequent stage that has already been generated by the prefix network is checked and the correction value (1010) is added. The final set of multiplexers select the output based on the operation selected. The only thing that needs to be made sure in this logic is that during subtraction the subtrahend is always the smaller number (in magnitude).In the proposed architecture this is managed by using the outputs of the (P*, G*) blocks which have comparators in their logic.

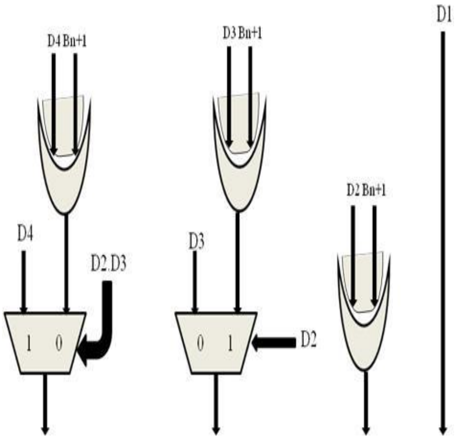

Figure 7: Correction Block for BCD Subtractor

Thus essentially there is no additional latency overhead due to this comparison. As it comes to the binary operation of the architecture, the prefix network is the major block. The pre-computation of the binary numbers consists of an array of XOR gates which compute the 2’s complement when subtraction needs to be performed. Then the propagate and generate bits are generated based on the equations mentioned in the previous section. These bits are sent to the prefix network which generates the carry at every stage.

These individual carry bits are XORed to the corresponding propagate bits for every stage to compute the sum/difference. Thus the prefix network that is the major block both in the BCD and binary computation is shared between the two operations to facilitate re-configurability. The signed numbers are taken care by the control logic at the beginning which takes the two sign bits and OpSelect (Operation Select) as inputs to compute the control signal that selects the appropriate multiplexers depending on the operation.

5. RESULTS AND DISCUSSION

The proposed architectures have been simulated in the simulation environment mentioned above and the results for latency and area for 32-bits architectures are given .Comparison of the existing architecture and the proposed architectures are in terms of latency and area.

DESIGN	LATENCY(ns)	AREA(LUTS)
Hwang	10.5	158
Fisher	10	584
Humberto	12.1	495
Proposed	8.3	512

6. CONCLUSION

Existing and proposed architectures for the BCD and binary reconfigurable adders are presented, simulated and compared. A novel way of implementing subtraction in Binary Coded Decimal without the use of 10’s complement is explained.

Though there is the necessity for the check of magnitude of the two numbers before subtraction it is implemented in a way as not to affect the latency. All the cases where correction is necessary and the logic correction blocks The proposed BCD and binary adder/subtractor is at least 11% faster than the fastest one till now while occupying a considerable amount of area.

REFERENCES

[1] Draft IEEE Standard for Floating-Point Arithmetic. New York: IEEE, Inc., 2004,

[2] Michael F. Cowlishaw, “Decimal Floating Point: Algorithm for Computers,” Proceedings of the 16th IEEE Symposium on Computer Arithmetic (ARITH ’03), pp 104-111, June 2003.

[3] M.S. Schmookler and A.W. Weinderger, “High Speed Decimal Addition”, IEEE Transactions on. Computers, vol. C-20, pp. 862-867, August 1971.

[4] W. Bultmann, W. Haller, H. Wetter, and A. Worner, “Binary and Decimal Adder Unit," U.S. Patent #6,292,819, September 2001.

[5] R.D. Kenney and M.J. Schulte, “Multi operand Decimal Addition,” Proc. IEEE CS Ann. Symp. VLSI, pp. 251- 253, Feb. 2004.

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

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

[8] H. Fischer and W. Rohsaint. Circuit Arrangement for Adding or Subtracting Operands Coded in BCD-Code or Binary-Code. Siemens Aktienge sells chaft, US patent.