DESIGN AND IMPLEMENTATION OF CODE CONVERTERS FOR HIGH SPEED MULTIPLIER APPLICATIONS

P. Sai Lalitha Durgamba [1], K. Chaitanya [2]

[1]*M.tech,VLSI design, GITAM, Visakhapatnam*
[2]*Assistant Professor, Department of Electronics & Communication Engineering, GITAM, Visakhapatnam*

---***---

Abstract – *The aim of this project to design a code converter which can be applicable for high speed multiplications. Using shifters and adders, 8 and 16 bit BCD to binary converters are designed. A binary multiplier is also developed by using multiplexers which acts as base for 8-bit BCD multiplication for easy multiplication. The proposed 8-bit and 16-bit converters, along with 4-bit binary multiplier are placed in 'Urdhava Triyagbhyam' sutra based decimal multiplier. The simulations were carried out using Xilinx Vivado/ISE simulator and Cadence (45nm) technology library and implementation is done on Artix -7 FPGA platform. Results shows that simulation using Cadence (45nm) provides lesser delay.*

Key Words: 8-bit BCD to binary converter, 16-bit BCD to binary converter, binary multiplier, Concatenation.

1. INTRODUCTION

The necessity for decimal multiplication is increasing in financial and computer based applications. The decimal fractions provide more accurate results than any number system, priority for decimal arithmetic operations have raised. To interact with computer applications, there is need to convert decimal to binary and vice versa. The process of converting binary to decimal format is simple. To convert decimal format to binary is quite complex. In order to reduce the difficulty, binary coded decimal to binary conversion is used. However, the existing BCD to binary converters provide higher delay because of presence of multipliers. N.McDonald designed binary to BCD converter using shift algorithm with a delay 4.49ns [1]. In this project, BCD to binary converter with shift algorithm is designed in order to reduce delay further.

1.1 Proposed 8-bit BCD to binary converter

A similar way of using shift and add algorithm is used to design BCD to binary multiplier. The 2-digit BCD number is divided into lower nibble and upper nibble. Upper nibble is concatenated with 4-bit zeroes. The concatenation is must for the purpose of successful shifting. The concatenated value is shifted twice and shifted once. The twice shifted value and one shifted value are added with Lower nibble value.

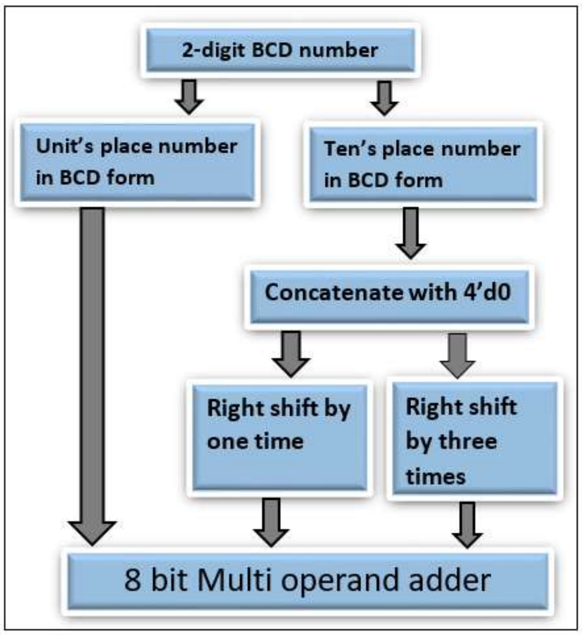

2-digit BCD number
Unit's place number in BCD form
Ten's place number in BCD form
Concatenate with 4'd0
Right shift by one time
Right shift by three times
8 bit Multi operand adder

Fig- 1 8-bit BCD to binary converter

1.2 Proposed 16-bit BCD to binary converter

Input BCD value of 4-digit number is sliced into two halves where each half represents 8-bit. With the help of 8-bit BCD to binary converter, each 2-digit number is fed to each converter which results two binary outputs. One binary output is concatenated with 8-bit zeroes and shifting is done. During right shifting, information may lost. In order to not to lose, Concatenation should be done. Later, all the values are added by 16-bit multioperand adder as shown in fig-2.

For N-bit conversion, N/2 bits are used for concatenation purpose and N-bit adder is proposed.

2. PROPOSED BINARY MULTIPLIER

A simple 4-bit binary multiplier output is generated by more adders. An effective designed of binary multiplier is designed so that only one simple adder can be used. For this purpose, multiplexers came into existence in this multiplier. However, only one input is given as income to mux whereas other input acts as select line to mux. Input of 1-bit is acted as select line to each multiplexer as shown in fig-3. This proposed multiplier is attached with binary to BCD converter for the purpose of conversion.

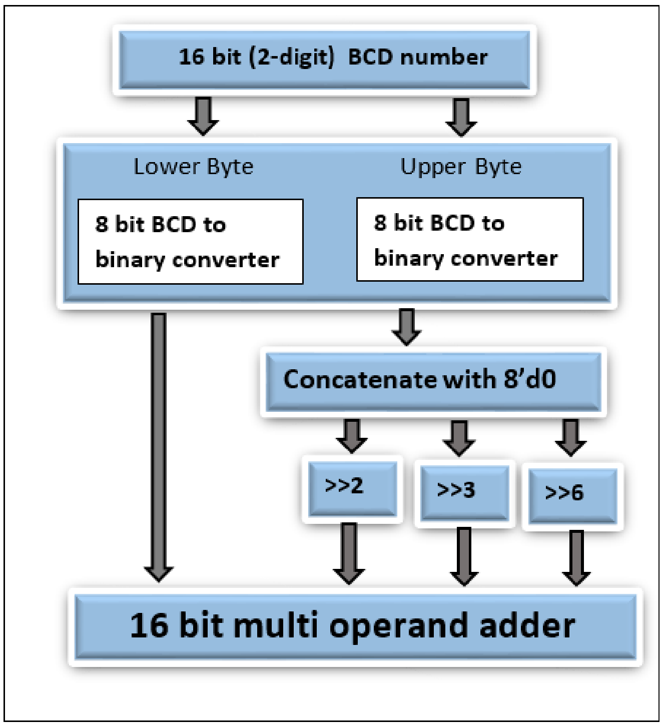

16 bit (2-digit) BCD number
Lower Byte
Upper Byte
8 bit BCD to binary converter
8 bit BCD to binary converter
Concatenate with 8'd0
>>2
>>3
>>6
16 bit multi operand adder

Fig-2 Proposed 16-bit converter

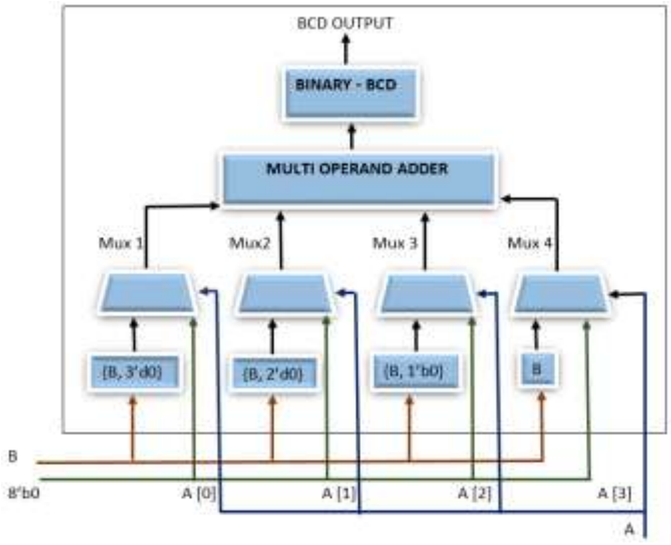


Fig-3 Proposed binary multiplier

3. DECIMAL MULTIPLICATION

The below fig-4 represents architecture for decimal multiplication. The decimal inputs are noted as multiplicand and multiplier. The combination of multiplicand and multiplier acts as one multiplication. The arrangement of inputs are based on 'urdhava Triyagbhyam' sutra. Inputs are arranged vertical and cross-wise method as per sutra. Each 4-bit multiplication uses 4-bit binary multiplier as base. The partial products obtained are added by BCD adders and final binary output can be obtained by using Proposed BCD to binary converter. Same architecture can be applied for N-bit decimal multiplication. It can extend from 8-bit to 64-bit multiplication.

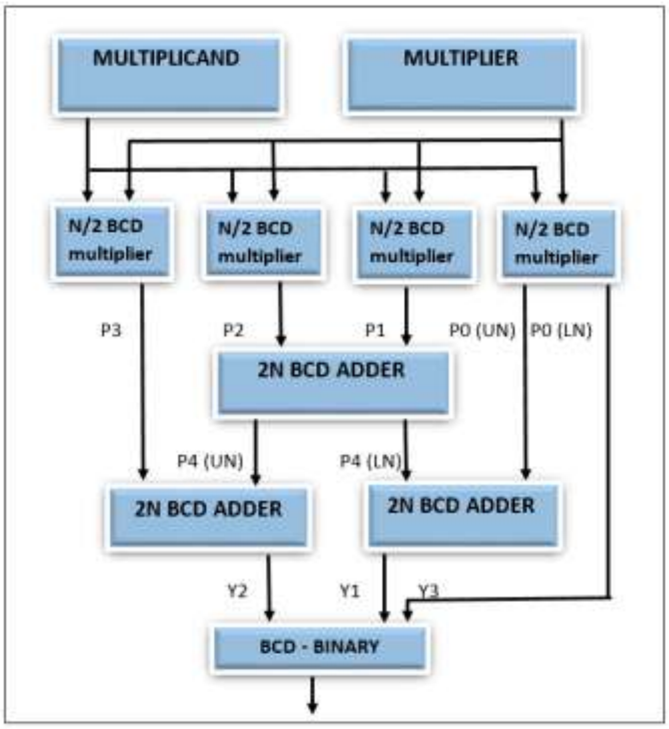


Fig-4 Proposed decimal multiplier

4. RESULTS AND DISCUSSION

The simulations were carried out using vivado and Cadence (45nm) technology.

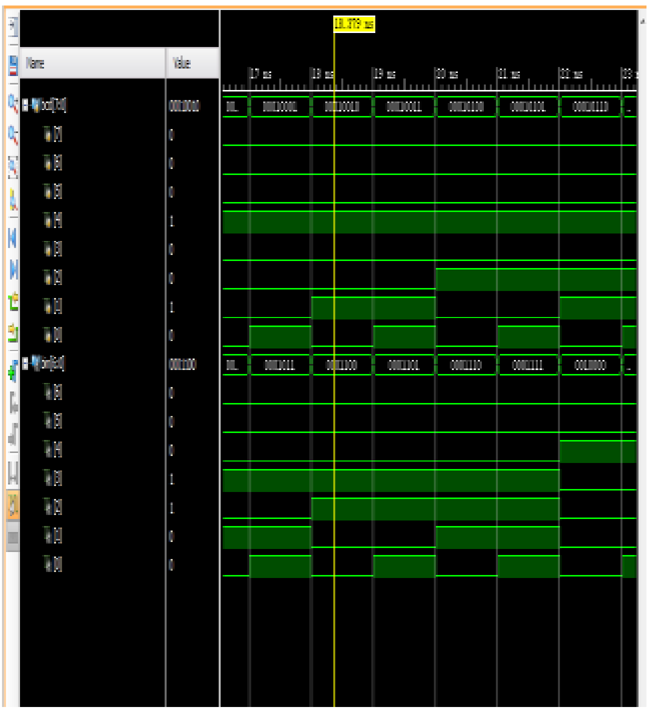


Fig-5 Simulation for 8-bit conversion

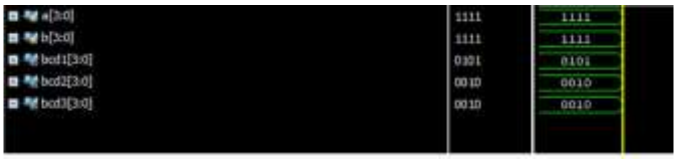


Fig-6 binary multiplier

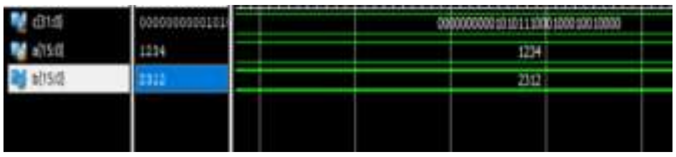


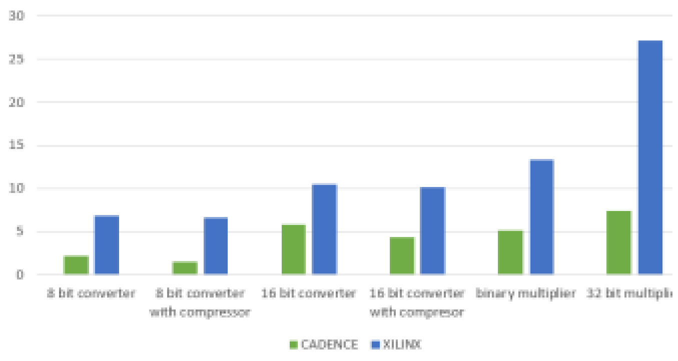
Fig-7 decimal multiplier

Comparison of 8-bit and 16-bit converters calculations are in the below table-1.

Table -1: power, area and delay calculations.

Calculations for proposed architectures			
architectures	delay	area	Power
8-bit BCD to binary converter	2.20ns	44	866nw
16-bit BCD to binary converter	1.42ns	44.4	879nw
Binary multiplier	5.8ns	91.3	1589nw
Decimal multiplier	4.52ns	21.7	1789nw

Chart -1: Comparisons of simulations



6. CONCLUSIONS

Finally, BCD multiplier is designed with the help of proposed BCD to binary converters and BCD adders. The inputs to the BCD multiplier are decimal numbers which are given to $n/2$ BCD multipliers. So, for input of 32 bits, each BCD multiplier takes 8 bits each which are again divided into two numbers. Then, the partial products along with appending zeroes are fed to "2n" BCD adders until the arrival of BCD output. At last, BCD output produced is fed to propose BCD to binary converter to take the binary output.

REFERENCES

[1] N. McDonald, Binary to BCD Conversion Algorithm,
[2] IBM Corporation, IBM Power6, [Accessed:02-02-2016]
[3] T. Lang, A. Nannarelli, A radix-10 combinational multiplier, Proceedings of Fortieth Asilomar Conference on Signals, Systems and Computers(ACSSC), IEEE, 2006, pp. 313–317.
[4] R.K. James, K.P. Jacob, S. Sasi, Performance analysis of double digit decimal multiplier on various fpga logic families, Proceedings of 5th Southern Conference on Programmable Logic(SPL), IEEE, 2009, pp. 165–170.