

# Implementation of Vedic Multiplier and Floating Point Matrix Multiplier in Image Compression Applications

T. Amy Prasanna

**ABSTRACT:** It is implemented into low Power Verilog Architecture to the area for digital images Process application, In the matrix multiplications are one of the key arithmetically operations. And the constructed into VLSI architecture for Low Power, High Speed & Lowarea, Matrices Multiplications designed into become rare. In the projects, is a simple work of fiction in Verilog architectures with Floating point matrix multiplier be presents. The designs into consider as Pseudo codes with the matrix multiplications, CSD multiplication algorithms with power reductions, Convention floating points as number formatting & Pipeline concept with as improves speeds. In the Floating point matrix multiplier design as appropriate with anyone arbitrary sizes of the matrix among the followed matrices rule. It is designed may also gives as higher precision outputs. The simulation result is perfect matched into the MATLAB result.

**Keywords:** Floating Point Matrix Multiplier (FPMM), Canonic Signed Digit (CSD).

## I. INTRODUCTION

The model as the electronic, in the field in VLSI, The digital image Processes & relating field in that has been vast demanding. Therefore the designed in VLSI architecture have been done. The most care to reach the demanding design constraint like as lowpower, LowArea & HighSpeed. Furthermore, as the effect designing in the VLSI architecture form Floating point matrix multiplier as required into DIP application and has decided in the systems performances. Normally it is involves for complex computations. Several methods in developing into review at purposed, except them may not simple. The paper is to simple work of fiction VLSI architectures with Floating point matrix multiplier at developed. In this project are organizing at following, sections 2 at presented as the proposes as floating point matrixmultiplier, section 3 shown in the experimental result down for the

MATLAB result into proposing works, & finally conclude that future scopes is present into sections 4.

## II. LITERATURESURVEY

It is designing as MatrixMultiplier is done with the Fixed Point cases. [7]have been designs at a fixed point matrixmultiplier so the matrix sizes as increased into bandwidth moreover increased in while it is limited in the scalable of the hardware resource. [8] have proposing ,model as algorithms into which is sizes as the memories needed into

direct proportional into sizes as the matrix & it is need a fix bandwidth. [9] was present an systolic architectures as we may dealing into multiple PE as a time become and required most hardware resource.[10], [11] consider into architectures because the requires efficiency are not meet.

## III. PROPOSED FLOATING POINT MATRIX MULTIPLIER

The judge at images is compress & dividing into subimages as definite sizes in this example 3X3. We are load into the file (say 'b') follows as represented in sub images within matrices into pixel value. Nowadays considered in Bias matrix as sizes into efficiently use CSD [1-4] multiplication algorithms. Most as about CSD multiplications algorithms can be discuss in the following section. This is values represented BCSD (Binary CSD) & load in these are another files (say 'a'). Files as 'a' considers on Multiplicand matrix & File 'b' to consider on Multiplier matrix & both is represents into binary value.

In this two file is takes as to BRAM's or DRAM's among the Xilinx tools uses Verilog HDL/VHDL. Pseudo codes as Matrices Multiplications [5] as chased with multiply Multiplication Matrix & Multiplier Matrix. Most of the Pseudo code form matrices as multiplication can be discuss into the follow sections. Useful as these are codes in first rows as the first matrix & one/one column in second matrix is multiplied on accumulated is produced 1st row as the result matrices. The procedures are repeated into all rows into 1st matrices for the yields remains as row into result matrices. Therefore as consider 8 bit with the precisions value at floating point value in the result matrix. Which is multiply as consider real value is partially at product of the resultant matrices is now such as binary value as them into shift for 8 position left 8 to the assumptions as precision width. Therefore we shift backs within 8 position that is 8 bit Right Shift & dividing into 256.0 ( $2^8$ ) at compensated in earliest shifts & results real value & them we load into other files known as 'c'. But CSD conversions are not finished earliest as stored into BRAM and DRAM, it is finish within pipeline mechanisms to discussed into the follow section.

**Revised Manuscript Received on December 30, 2019.**

**T. Amy Prasanna**, Professor, Department of Electrical and Electronics Engineering, Malla Reddy College Of Engineering for Women, Telangana State, India. Email: drsyamalasisivaprasad@gmail.com

# Implementation of Vedic Multiplier and Floating Point Matrix Multiplier in Image Compression Applications

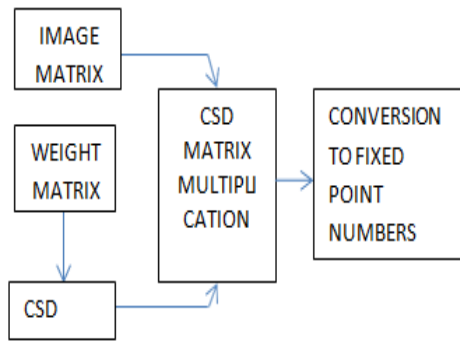


Figure.1. floating point matrix multiplier

- These are CSD no's contain into minimum numbers as non-zero bits, hence the name canonic.
- The CSD representation is unique for a given number.
- CSD numbers cover the range  $(-4/3, 4/3)$ , in which the values in the range  $[-1, 1)$  are of great interest.
- The number of nonzero bits in the range of  $[-1, 1)$  for W-bit CSD numbers is
- Therefore CSD number contains around 33% fewer nonzero bits than normal numbers.

The algorithms as a converted into binary numbers as CSD numbers as presents into below. This binary A as represents into  $A = a_{w-1}a_{w-2} \dots a_1a_0$  & it has been CSD number as

$$C = c_{w-1}c_{w-2} \dots c_1c_0.$$

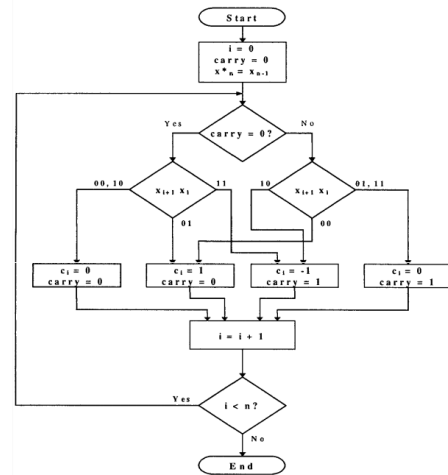
```

for (i= 0 to W-1)
{
     $Q_i = a_i \text{ xor } a_{i-1}$ 
yi
= yi-
1
and

```

$$y_i c_i = 1 - t_i - t_i$$

Constant multiplication may be done among the subtracts and added into partial product corresponds into position in the non-zero bits in the constant multiplier. A CSD coded multiplier contains minimum number of nonzero bits. Then it is required.



## IV. REVIEW OF CSDALGORITHM

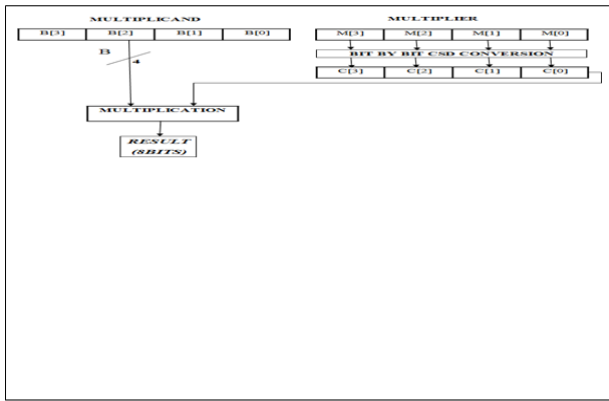
In the numbers into addition operation is require as now into one lesser than as the number into non-zero bit into the constant numbers. At reduces in the numbers in the power consumptions & areas, as the constant numbers may be code therefore (i.e), it contain minimum non-zero bit. It is to be done among the represents in the number of canonic signed digits form. These are characteristics CSD representation as The formulations as the algorithm flow charts, into while it is suitable form hardware implementations, as illustrate into above figures.

The algorithm converted into 2's complements to CSD. This is algorithm need as defined as auxiliary signal "carry" & replicates in the MSB as then n-bit binary numbers, as  $x_{n-1} x_{n-2} \dots x_1 x_0$ , in the bit position  $x_n$  on algorithms form computes into the CSD formats for W-bit number to presents. The Denotes in the 2's complements represented as the number A as  $A = a_{w-1} a_{w-2} \dots a_1 a_0$  & it is CSD represented into the numbers as  $A = a_{w-1} a_{w-2} \dots a_1 a_0$ , as the conversions as illustrate used as the follow as iterative algorithms.

## V. CSD VEDIC AND FLOATING POINT MATRIX MULTIPLICATION

Multiplicative as the primitively arithmetic operations as involves into that work & the same may be decides into performances as the entire designs. Therefore we has been consider as CSD multiplications algorithm into which partially product are lesser because in that power dissipations as reduces & speed into computation as improves. As briefly literature as it is discussing about below paragraphs. Form as illustration size as both multipliers & multiplication and is taking into 4-bit. At 4-bit multipliers are converts as 8-bit CSD number 2 bit into every bits into multiplier among the Bit as Bit CSD converted modules & immediately sending as to multiplications blocks on show into Figure.1a [3]. Internally processor as with the blocked into multiplication as the Figure.1a as show as Figure 1b. CSD for Multiplier contain '00' and '10' and '11' representing 0 and +1 and -1 respectively. Which is '10' and '11' result are addition





Results & multiplicand shifted through the count values. It repeats as until last bits. Finally products can be store into results.

### VI. RESULTS AND DISCUSSIONS

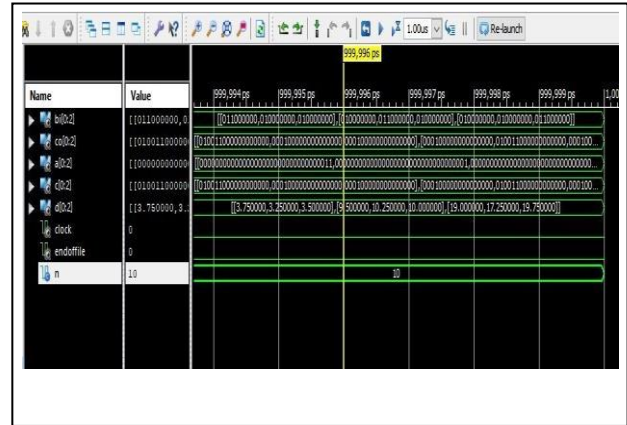
Among in the matrices multiplications may be extend form larger size as matrix if we considers in the size as 3X3 form towards multipliers & multiplicand matrices & precision into 8 bit position. Simulation result as the FPMM as show as into Figure.4 (d) form the Multiplier Matrix into Figure.4 (b) & MATLAB Results:

a =			
	3	1	2
	4	7	6
	12	5	15

Fig.3 (a) Multiplicand

b =		
0.75	0.50	0.50
0.50	0.75	0.50
0.50	0.75	0.75
3.7500	3.250	
3.5000	9.5000	
10.2500	10.0000	
19.0000	17.2500	
19.7500		

Fig.4 (b): Multiplicand matrix



### VII. CONCLUSION AND FUTURE SCOPE

In simple writer architectures on Floating Point Matrices Multipliers are proposed, as Simulated & Verified for MATLAB result. It has been very used as digital images compressions application as involved into serious intellectual messages.

Now, on simple writer architectures are proposing model for simulations, if it is extend form synthesis therefore as this may useful as implement on FPGA.

### REFERENCES

1. Keshab K.Parhi,"VLSI DIGITAL SIGNAL PROCESSING SYSTEMS:DESIGN AND IMPLEMENTATION", John Wiley & sons Publishing Company Inc.,1999.
2. Shoab Ahmed Khan," Multiplier-less Multiplication by Constants," in Digital Design of sign preparing frameworks: A commonsense methodology, John Wiley and children Publishing Company,2010,pp.253-299.
3. M.Lakshmi Kiran and K.Venkata Ramanaiah," Implementation of High Speed and Low Area Digital Radix-2 CSD utilizing Pipeline Concept", Internation3al Journal of Electronics and Communication Engineering, Vol.10, No.1, pp.53-61.
4. M.Lakshmi Kiran and K.V.Ramanaiah," Design and Implementation of High Speed Radix-2 CSD based Floating Point Multiplier", International Journal of logical research in multidisciplinary thinks about. Vol.3, Issue.7, pp.17-22
5. "Verilog Code for Matrix Multiplication"- for 2 by 2 Matrices. (2015, November 18). <http://verilogcodes.blogspot.in/2015/11/verilog-code-for-lattice-multiplication.html>
6. Sangjin Hong, Kyoung-Su Park and Jun-Hee Mun, "Structure and Implementation of High
7. Amira, A., Bouridane, P.Milligan and P.Sage, 2000. "A high throughput FPGA execution of somewhat level framework item," Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, August 8-11, 2000, Lansing, Michigan,pp-396-399.
8. Jang, J.W., S.B. Choi, V.K.Prasanna, 2005. "Vitality and time effective grid increase on FPGAs," IEEE Transactions on Very Large Scale Integration Systems VOL. 13, NO. 11, APRIL2005.
9. Campbell, Scott J. also, Sunil P.Khatri. "Asset and postpone productive grid duplication utilizing fresher FPGA gadgets" ACM Great Lakes Symposium on VLSI.