Performance Boosting Components of Vedic DSP Processor

Anuradha Savadi
Electronics and communication engineering
PDA college of Engineering
Kalaburgi, Karnataka, India

Raju Yanamshetti
Electronics and communication engineering
PDA college of Engineering
Kalaburgi, Karnataka, India

***Abstract*—** This paper deals with performance boosting of the DSP processor, by introducing the different algorithms in processor blocks ALU, MAC, Encoder/ decoder at. al. To in decrease the processing delay of Brent Kung adder is used, instead of other adder. Vedic sutras like Urdhava Tiryagbhyam and nikhilam sutras are used to increase the speed floating point multiplier, MAC and filters and other signal computations. The verilog HDL is used and the validated through extensive simulation. Synthesis results and attainment scrutiny of each systems components confirmed significant performance meliorism in the proffered DSP processor over the extant one.

Keywords- *vedic sutras; Brent kung adder; floating point multiplier; DMC encoder/decoder.*

*****

I. INTRODUCTION

In the present era the DSP processors are in highest demand to meet the requirement of fastest communication world. In this paper we are dealing with the design of high speed DSP processor with optimal area and power consumption. This made up of four stage pipeline architechture and vedic sutras to enhance the speed as it is the highest priority in a real world. In many DSP computations multiplier is the major element. In various PCs and diverse sort of processors, adders are used exclusively. Adders are used as s piece of multipliers, in DSP processors to execute diverse computations like FFT, Filter at.al. Hence floating point multiplier design with vedic sutra is used to increase the speed as well as accuracy of the parameter. In the proposed work we are discussing very few blocks which enhances the performance like Encoding/Decoding , multipliers.

II. PROPOSED WORK

For the proposed vedic DSP processor decimal matrix control(DMC) encoding/decoding methods are used for any kind of data like text,image and video at.al. proposed work deals with high speed floating point multiplier with vedic algorithms with a IEEE 754 floating standard.

A. Encoding/Decoding

DMC alogorithm is used in the proposed Processor which enhances error detection and correction capabilities.

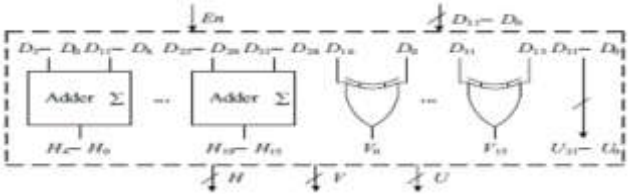


Figure 1: Block diagram of 32-bit DMC encoder structure

A 32-bit (d) data is fed to encoder, from which horizontal (h) and vertical (v) check bits are computed as follows:

h[4:0] = d[3:0] + d[11:8];
h[9:5] = d[7:4] + d[15:12];
h[14:10]= d[19:16] +d[27:24];
h[19:15]= d[23:20]+d[31:28];
v[0] = d[0] ^ d[16];
v[1] = d[1] ^ d[17]; and so on…

h and v bits are stored in SRAM. The decoder corrects these MCUs with the help of syndrome bits (s). The decoder uses horizontal syndrome (hd) bits to detect errors and uses vertical syndrome (vd) bits to correct those errors.

Decoding
s[0] = v[0] ^ vd[0];
s[1] = v[1] ^ vd[1];
s[2] = v[2] ^ vd[2]; so on..

Detection
h0h4 = hd[4:0] - h[4:0];
h5h9 = hd[9:5] - h[9:5];
h10h14 = hd[14:10] - h[14:10];
h15h19 = hd[19:15] - h[19:15];

Correction
D0 corrected= d0^s[0];

DMC computation for error detection/correction figure 2 symbolic representation is shown.

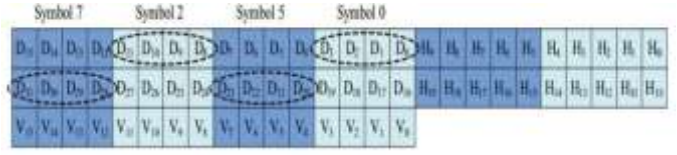
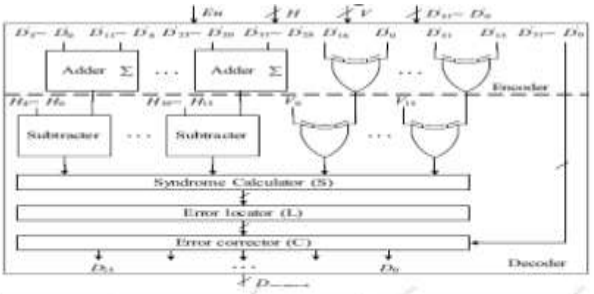


Figure 2: logical representation of 32-bit DMC code

 ISSN: 2321-8169
960 – 964



Extra circuit	En signal		Function
	Read signal	Write signal	
Encoder	0	1	Encoding
	1	0	Compute syndrome bits

Figure 3 : 32-bit DMC decoder.

B. Floating point multiplier

Floating point number system represents as shown in figure 4, it consist of four units namely sign calculation, formalizer, exponent and mantissa calculation units.

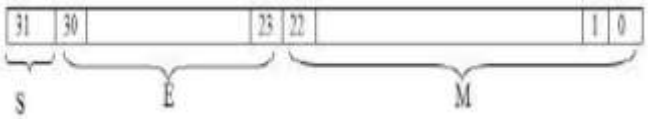


Figure 4: IEEE754 floating point number format

Where S= sign bit, M=mantissa bits and E=exponent bits.

Computational algorithm is as follows:

Step1: Multiplication of significand i.e. $(1.M_1 * 1.M_2)$
Step2:Placing the decimal point in the result.
Step3:Addition of the exponent i.e. $(E_1 + E_2 - bias)$.
Step4:Obtaining the sign bit i.e. $(S_1 \; XOR \; S_2)$
Step5:Normalizing the result i.e. obtaining 1 at MSB of the results significand.
Step6: Rounding the result to fit in available bits
Step7: Checking for overflow /under flow

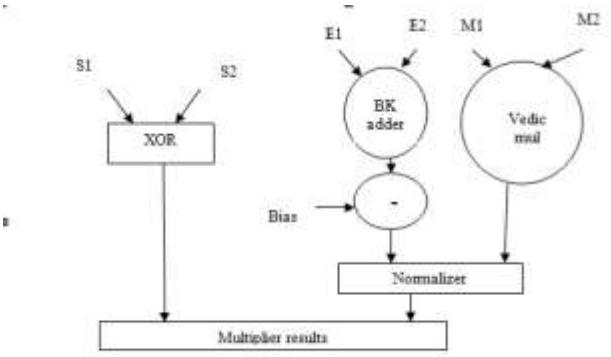


Figure 5 : Block diagram of proposed floating point multiplier

The concert of Mantissa calculation is the foremost component of the multiplier, which is designed by urdhva-tiryakbhyam sutra with vertical crosswise multiplication, is used for owning better speed and power reduction. The working of this vedic multiplier is as shown in figure 6. The adder used in this multiplier is Brent Kung adder which has less propogation delay compared to the extant one. From this tree structured adder computational speed will be more with less area utilization.

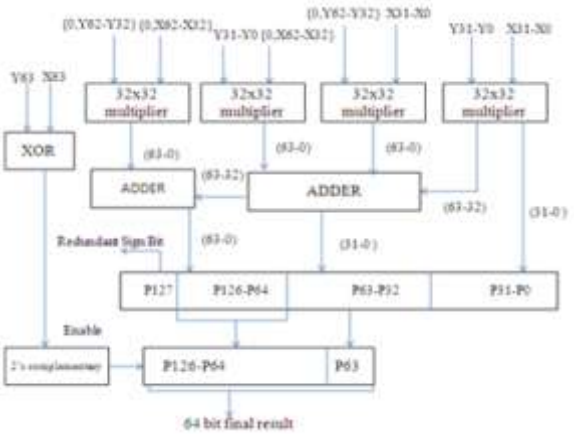


Figure 6: Block diagram of urdhva-triyakbhyam multiplier.

The exponent calculation unit can also be designed by Brent Kung (BK) adder (Figure 5), is a pretty known logarithmic adder structural design, which one of the parallel prefix adder. BK adder will give the optimal stages to propogate from input to all output stages. Pre-processing stage, carry generation network and post processing stages are the three stages of parallel prefix adder , computational equations are as follows:

Pre-processing stage:

Propagate signal= Pi= Ai ^ Bi

Generate signal= Gi=Ai & Bi

carry generation network (figure 7):

$CP_{i:j}=P_{i:k+1}$ & Pk:j

$CG_{i:j}=G_{i:k+1} | (P_{i:k+1}\&G_{k:j})$

Post processing stage:

Ci-1=(Pi & Cin)| Gi

$Si=Pi \wedge C_{i-1}$

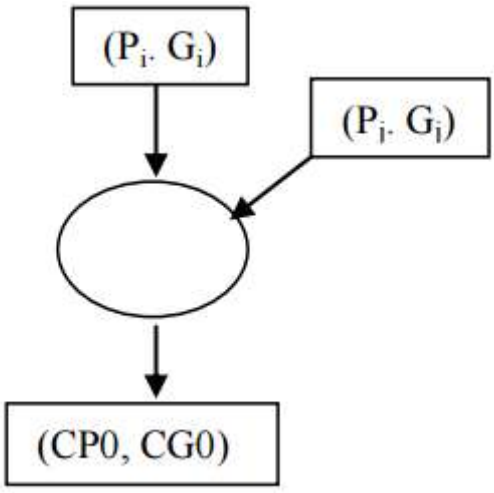


Figure 7 : carry generation network.

Working diagram of 4 bit Brent Kung adder is shown in firgure8, and same flow will be used for 32-bit, with color

 ISSN: 2321-8169
960 – 964

coding tree structure as black, grey and white. This adder will give less complexity, cost and increased arithmetic computations.

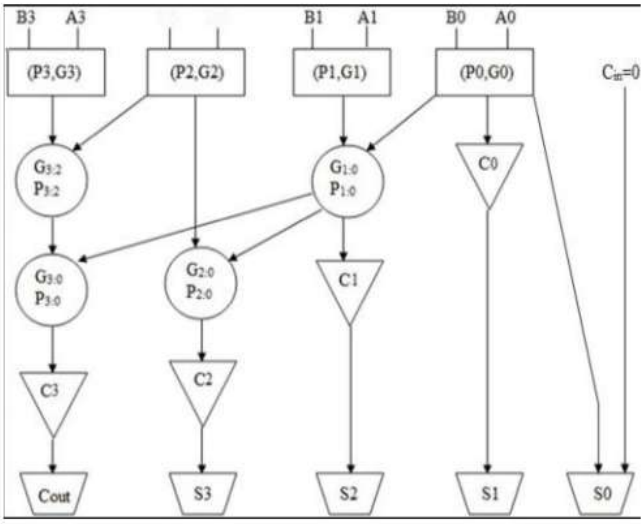


Figure 8: 4-bit Brent Kung adder

III. Results And Discussions

In this section will dicsuss the outcome of few major blocks of DSP processor which will help to boost the performance.

A. *Encoding/Decoding*

DMC encoding algorithm is implemented by encoder and decoder block as proven in RTL schematic shown in figure. 9

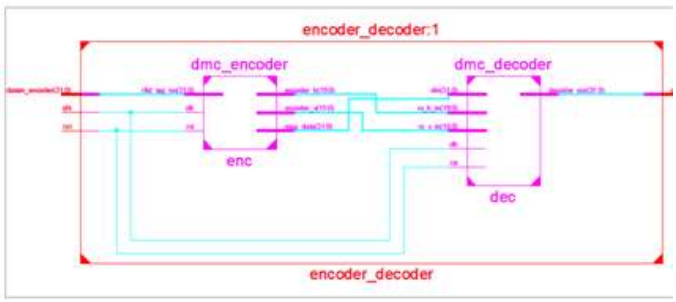


Figure 9: RTL schematic of DMC algorithm

The decoder block mainly consisting of decoing, detecting and correcting sub-blocks as shown in figure 10

Figure 10: RTL schematic of DMC decoder

Figure 11 give the simulation results of encoder and decoder with same data stream 10101011110011010001001000011

.

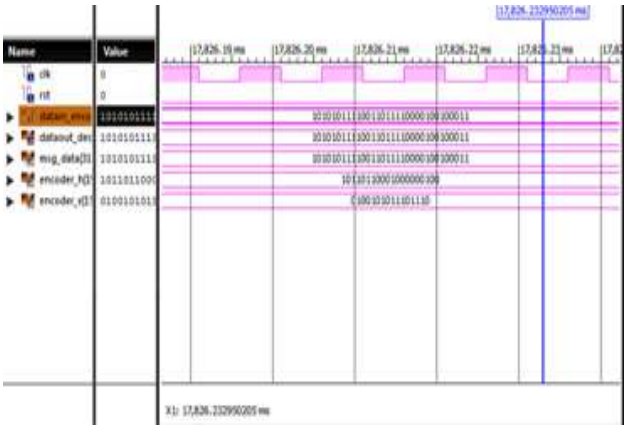


Figure 11: Simulation results for DMC

B. *Floating point multiplier*

Floating point multiplierconsist of Brent kung adder, vedic multiplier as major components. Figure 12 explains the area utilization of Brent kung adder which found to be less compared to the existing one.

```
Device utilization summary:
---------------------------

Selected Device : 5vfx70tff1136-1


Slice Logic Utilization:
 Number of Slice LUTs:                   59  out of  44800     0%
    Number used as Logic:                59  out of  44800     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     59
   Number with an unused Flip Flop:      59  out of     59   100%
   Number with an unused LUT:             0  out of     59     0%
   Number of fully used LUT-FF pairs:     0  out of     59     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                          98
 Number of bonded IOBs:                  98  out of    640    15%
```

Figure12 : Device utilization summary

Table1 shows comparision of performance parameters of existing adder with Brent Kung adder found to be 3.25ns which is more optimal.

Table 1:comparison of delays for different adders

Adders	Delays(ns)
Ripple carry adder	20.39
Carry look ahead adder	18.64
Kogge stone adder	16.5
Brent Kung adder	3.25

Figure13 shows the RTL schematic of 32-bit brent kung adder. By seeing the synthesis report it allows us to see RTL level

represent of HDL optimized for architecture,which helps us to resolve design process.

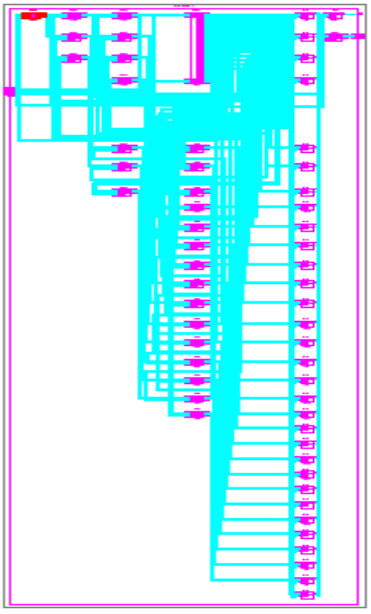


Figure 13 RTL schematic of Brent Kung Adder

Figure14 shows the simulation results of brent kung adder where inputs are A=10101011110011010001001000110100 and B= 00010010001101001010101111001101 and Cin= 0 and the output for this adder is Cout =0 and Sum= 10111110000000011011111000000001.

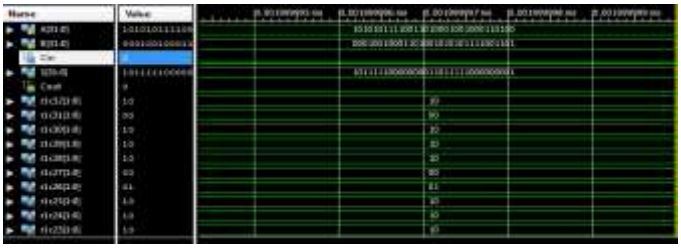


Figure14 Simulation result of brent kung adder

Figure 15 shows the resource utilization of floating point mulitiplier which is compact as compared to the exsisting mulitiplier with 15.167ns propogation delay.

```
Device utilization summary:
---------------------------

Selected Device : 5vfx70tff1136-1


Slice Logic Utilization:
 Number of Slice LUTs:                 7240  out of  44800    16%
    Number used as Logic:              7240  out of  44800    16%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:   7240
   Number with an unused Flip Flop:    7240  out of   7240   100%
   Number with an unused LUT:             0  out of   7240     0%
   Number of fully used LUT-FF pairs:     0  out of   7240     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                         203
 Number of bonded IOBs:                 203  out of    640    31%
```

Figure 15 area optimization of a floating point mulitiplier.

The figure 16 shows the simulation results of 64-bit floating point mulitiplier.

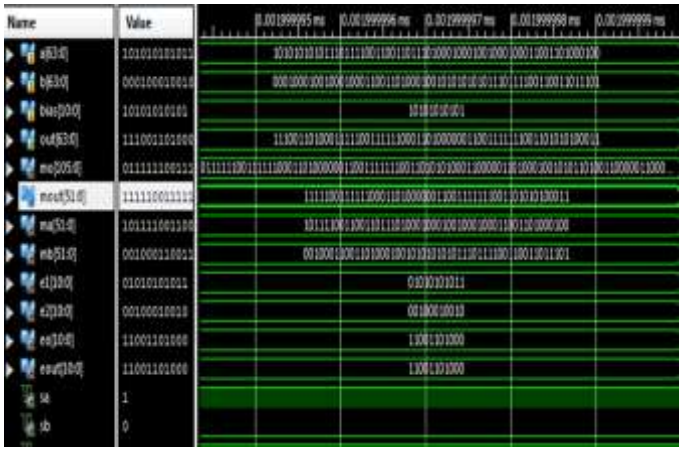


Figure16 :Simulation result offloating point multiplier

IV. CONCLUSION

In this work , performance parameters of DSP processor has been enchaned, by optimizing the sub-blocks like, Adders, Floating point multiplier, DMC encoder and decoder. All the results are verified and found comaritively better with the existing one. The overall speed of proposed processor is 212MHz, with a better throughput 229.8MB/s less power consumption and area.

ACKNOWLEDGMENT

We are very much thankful to the technical experts of PDA college of engineering and APPA institute of engineering and technology, for their continue support for this work.

REFERENCES

[1] Mohamed Al-Ashrafy, Ashraf Salem, Wagdy Anis, An Efficient Implementationof Floating Point Multiplier, 978-1-4577-0069-9/11/2011 IEEE.

[2] Maruti L Doddamani, Mala L.M, Design and implementation of an optimized floating point matrix multiplier based on FPGA, The Internation journal on science and technolegde, vol.2, may 2014.

[3] 3.P.Nithin,N.UdayaKumar and K. Bala SindhuriIndian Journal of Science and Technology, Vol 9(44),DOI:10.17485/ijst/2016/v9i44/101948, November 2016.

[4] Pallavi Saxena, Design of low power and high speed carry select adder using Brent Kung adder, International IEEE conference on VLSI Systems, Architecture, Technology and Applications 2015.

[5] Tasnim Ferdous, “Design and FPGA-based Implementation of a High performance 32-bit DSP processor” computer and information technology,ICCIT, IEEE conference, pp.484-489, Dec 2012.

[6] Sandesh S Saokar, R.M.Banakar, and Saroja Siddamal, "High speed signed multiplier for digital signal processing applications" signal processing , computing and control(ISPCC) 2012, IEEE conference, pp.1-6, March 2012.

[7] Surabhi Jain, Mukul Pancholi, Harsh Garg and Sandeep Saini, " Binary division algorithm and high speed Deconvolution algorithm", Electrical Engineering/Electronics, computer, telecommunications andinformation technology(ECTICON), IEEE conference pp. 1-5, May 2014.

[8] Mohamed Al-Ashrafy, Ashraf Salem, Wagdy Anis, "*An Efficient Implementation of Floating Point Multiplier*", 978-1-4577-0069-9/11/$26.00 ©2011 IEEE

[9] Korra Tulasi Bai, J. E. N. Abhilash, "*A New Novel Low Power Floating Point Multiplier Implementation Using Vedic Multiplication Techniques*" ,International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 3, Issue 4, Jul-Aug 2013.