A LOW-POWER HIGH-SPEED APPROXIMATE MULTIPLIER FOR MAC APPLICATIONS

M.N. Bhavana[1], J. Prasad Babu[2].
PG Scholar[1], Dept of ECE (VLSI), SIETK, Puttur, India
Associate Proffesor[2], Dept of ECE, SIETK, Puttur, India

ABSTRACT: MAC unit performs both multiply and addition capabilities. This is frequently demanded in Digital Signal Processing applications. The multiply- accumulate (MAC) unit operates in two ranges, Firstly it calculates the manufactured from given numbers and feed the end result to the following stage of operation i.E. Addition or called as accumulation. If both the ranges are carried out in a single spherical then it is called fused multiply- accumulate (fMAC) unit. A lot of researches had been completed and performing on MAC implementation. In this paper we gift a brand new design of approximate multiplier that is utilized in MAC to reduce the hardware complexity of the Design. Moreover, the proposed multipliers are compared with exact multipliers, outperforming in performance parameters like area, delay and power consumption, for similar error values. The simulation results are shown in Cadence EDA tool with 90nm technology libraries.

Key Words: DTMAC, DSP, ECSM, FPGA, MAC, VLSI.

I. INTRODUCTION

The multiply and accumulate (MAC) Unit emerges large quantity of Digital Signal Processing (DSP) applications. It additionally complements the sign processing capability to the microcontroller for numerous programs inclusive of servo/audio control and so forth. Being an execution unit inside the processor, MAC implements a three-level pipelined arithmetic structure which optimizes eight×8 multipliers utilization. The proposed layout supports eight-bit operands. Any MAC unit supports mainly 3 functions those are: signed/unsigned operations, fixed-factor floating enter operations and Miscellaneous sign up operations

A MAC unit is usually manufactured from an accumulator adder and multiplier. Typically a carry pick or a deliver keep adders are used in most excessive cases used because of the DSP application requires of speedy overall performance The memory unit fetches the inputs from its memory vicinity and offers to multiplier for further multiply-accumulate operations. The final end result of MAC unit is stored at a MAC reminiscence place. The function of MAC demands that this entire procedure must be executed in a single clock cycle time [1].

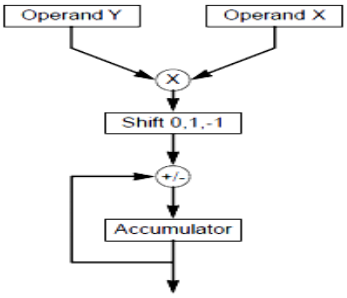


Figure A: Block diagram of MAC unit

The Goal indoors developing MAC unit is grasping from essential multiplier. We may want to make reference that MAC unit is a decorate version of primary multiplier unit that is applied in at most all microprocessors. It empowers superb scope of advanced digital signal processing (DSP) programs with constrained time of execution that is have confined range of clock cycles as consistent with the call for of application. Like few of the filters can adjust a do away with subsequently at some stage in the execution besides the algorithms which includes symmetrical changes called orthogonal transforms, and so on. Requests exact time of execution/calculation which from time to time depends on the capability of a processor.

II. EARLIER WORK

For error tolerant applications we employ surmised registering which could reduce the layout multifaceted nature with the aid of the usage of improving the strength effectiveness and framework execution. This take a look at manages a bleeding thing plan gadget for the estimation of multipliers. The halfway crafted from the multiplier is modified to present various open door states desire making functionality unpredictability of output. Execution of multiplier fuses 3 degrees: duration of incomplete product, fractional gadgets rebate is shifted for the accumulation of changed halfway devices depending on their chance. Directly right here we're using the superb country of a 8-bit multiplier with 8×8 fractional devices characterized below.

A. Approximate Tree Compressor

In fig 1(a), an accurate half adder is presented and the following equation can be obtained

$$\{c, s\} = a + b = 2c + s = [c + s] + c,$$

Here, $\boldsymbol{c} = \boldsymbol{a}\,\&\,\boldsymbol{b}$ and $\boldsymbol{s} = \boldsymbol{a} \oplus \boldsymbol{b}$, thus $(\boldsymbol{c} + \boldsymbol{s}) = \mathbf{a}|\mathbf{b}$. Basically based on the above conditions, by considering the essential logic block given in Fig. 1(b), for which the following equations may be acquired:

$$p = c + s,$$

$$q = c,$$

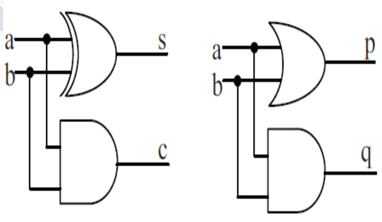
$$\{c, s\} = a + b = p + q.$$

This is called as an inexact adder cell (iCAC). Table I indicates sensible tables for an proper half adder and an iCAC. Note that the bit characteristic of c and that of s, p, and q are unique. As can be visible, q is proportionate to c. While p isn't always drawn in the direction of s, the right all out can be acquired with the resource of way of which incorporates p and q, so the iCAC isn't an erroneous adder but part of a cautious adder.

By extending the above state of expressions to m-bits, the going with state of expressions may be gotten:

$$S = A \oplus B = P + Q.$$

Where, A, B, P, and Q are m-bit operands, the bits which correspond to a, b, p, and q, respectively. A group of eight iCACs, used for 8-bit iCAC, is shown in Fig.2. below



(a) (b)

Fig. 1 (a) Accurate half adder and
(b) Incomplete adder cell.

TABLE I. TRUTH TABLES FOR ACCURATE HALF ADDER AND INCOMPLETE ADDER CELL.

Inputs		Outputs			
		Accurate half adder		iCAC	
a	b	c	s	q	p
0	0	0	0	0	0
0	1	0	1	0	1
1	0	0	1	0	1
1	1	1	0	1	1

Two 8-bit inputs:

$$A = \{a7, a6, a5, a4, a3, a2, a1, a0\}$$

$$B = \{b7, b6, b5, b4, b3, b2, b1, b0\}$$

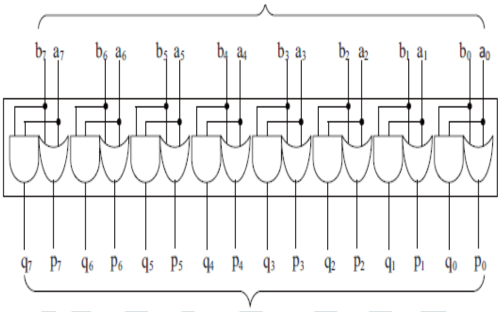


Fig. 2. A row of incomplete adder cells with two 8-bit inputs.

Two 8-bit outputs:

Approximate sum:

$$P = \{p7, p6, p5, p4, p3, p2, p1, p0\}$$

Error recovery vector:

$$Q = \{q7, q6, q5, q4, q3, q2, q1, q0\}$$

An ATC (Adder Tree cell) with n records assets is called an ATC-n, And the shape of an ATC with 8 records sources (ATC-eight) is tested in Fig. Three. The rectangular shapes building up segments of iCACs and the big affiliation of iCACs in every line (rectangular form) is depending upon the bit width of the facts property. For example, if there are eight m-bit inputs (D1, D2… D8), four sections of m-iCACs are required to fabricate a m-bit ATC-8. This redirection makes 4 evaluated aggregates, P1, P2, P3, and P4, and error restoration vectors, Q1, Q2, Q3, and Q4. Or alternatively gates produce the error correction vector V.As a last thing, the eight rows of information were decreased to 5.

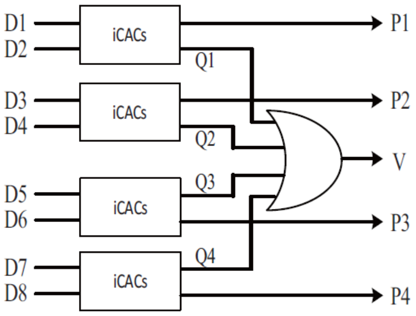
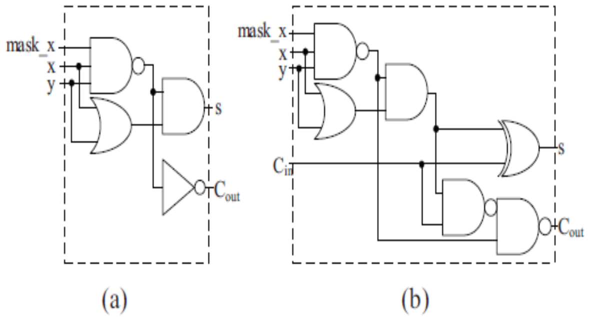


Fig.3. Structure of an approximate tree compressor with eight inputs.

B. Carry-maskable Adder

The proposed architectures of Carry-maskable half of adder and whole adder are proven in below Fig. Four. For the proposed half adder, while the given mask_x input is 0, S is x OR y and Cout is identical to 0. Otherwise if, mask_x is 1, then S is identical as accurate sum x XOR y and Cout is equal as accurate deliver x AND y.



(a) (b)

Fig.4. (a)Carry-maskable half adder, (b) Carry-maskable full adder.

C. Overall Structure

A n-bit Multiplier includes n-rows of partial products (PPs), so there are absolutely n × n PPs bits. The Partial Product Reduction is carried out in three tiers of operation (Stage 1, Stage 2, and Stage three) and the Carry Propagate Adder is executed at Stage 4.

In Stage 1, we will get 4 rows of partial merchandise (P1, P2, P3, and P4) and one errors correction vector (V1) via reducing or compressing the 8 rows of partial merchandise with the help of an ATC-eight. These 4 rows of include however reduced to 2 rows of partial products (P5 and P6) and some other blunders correction vector (V2) by the use of an ATC-four. A final level of iCACs are then method the generated partial merchandise P5 and P6, produces P7 and Q7. In exact, Stage 1 makes use of an ATC-eight, an ATC-four, and a phase of seven iCACs to reduce the 8×8 PPs to 4 rows of partial products (P7, V1, V2, and Q7).

In stage 2, there are 4 PPs for every one in each of bits 4 to 10. In order to acquire decrease route put off a excellent method is used that is, OR gates are implemented to add V1 and V2 which offers approximate sum. The white circles for V1 and V2 portrays the bits that are to be summed making use of OR logic gates. Basically Seven OR gates are required in total and the 4 rows are reduced to 3 rows.

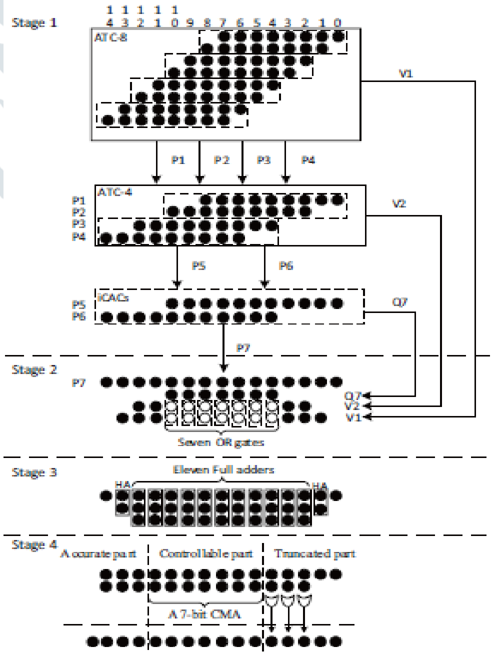


Fig.5. Structure of an 8-bit multiplier with 8×8 partial products.

In stage 3, half adders and complete adders are implemented to compress the 3 rows of reduced partial products into to 2 rows. We want 11 full adders for bit places 2 to 12 and half adders are required for bit places 1 and 13.

In diploma four, In order to reduce the propagation of bring the Carry Propagate Adder is splitted with the aid of three elements. Since the decrease bits are not a good deal substantial for accuracy element, so bits 0 to 4 are defined because the truncated element and 3 OR gates are used to generate the final result for bits 2, 3, and 4 consequently there may be no bring generated from the truncated part, and the length of the CPA is decreased to 10.

III. PROPOSED MAC

In most of digital signal processing (DSP) applications the pivotal tasks commonly include a few multiplications as well as summations. For timeframe signal process, a fast and high yield Multiplier-Accumulator (MAC) is regularly a key to understand an elite computerized signal process framework. Inside the past couple of years, the most idea of MAC style is to fortify its speed. This can be on the grounds that; speed and yield rate is regularly the need of advanced signal process framework. Aside from the age of private correspondence, low power style furthermore turns into another principle configuration thought.

This can be because of, battery vitality available for these transportable item confines the capacity utilization of the framework. Consequently, the most inspiration of this work is to inquire about various pipelined multiplier/ accumulator models entryway style procedures that are suitable for executing high turnout flag process calculations and at the indistinguishable time win low power utilization. a standard MAC unit comprises of (quick multiplier) multiplier and a accumulator that contains the include of the past continuous item. The function of the MAC unit is given by the resulting condition:

$$F = \sum Ai\ Bi$$

The objective of a DSP processor configuration is to enrich the speed of operation, so that we use MAC unit and on the same time lower the power admission. In a pipelined MAC circuit, the delay of pipeline level is the equal to the delay of a 1-bit full adder (Jou, Chen, Yang and Su, 1995). Evaluating this delay will help with making sense of the general delay of the pipelined MAC.

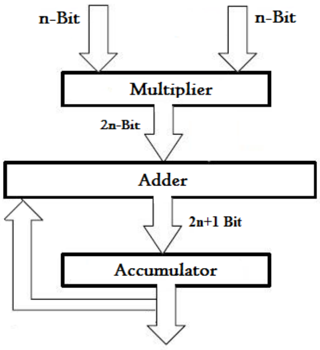


Figure 6: Basic structure of MAC

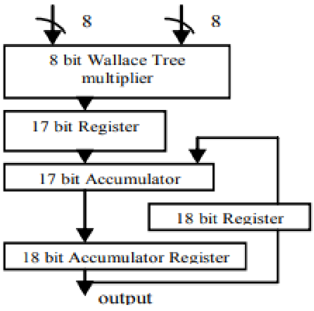
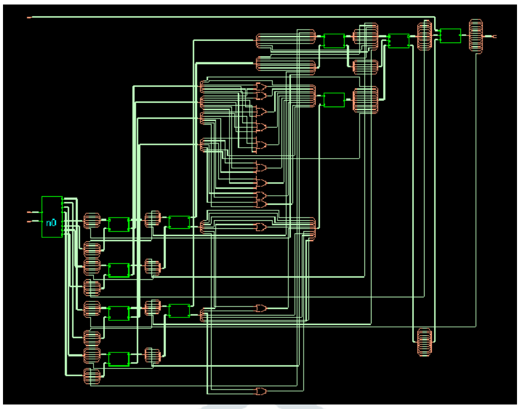


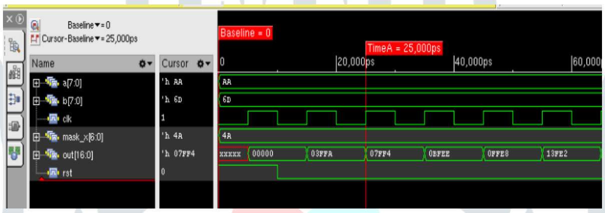
Figure 7: MAC architecture

IV. EXPERIMENTAL RESULTS

RTL schematic structure is shown in below:

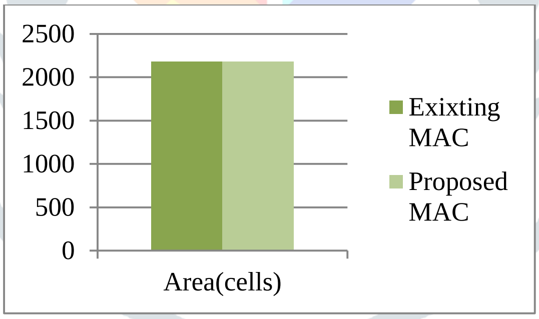


The simulation output for the proposed MAC is shown in below:

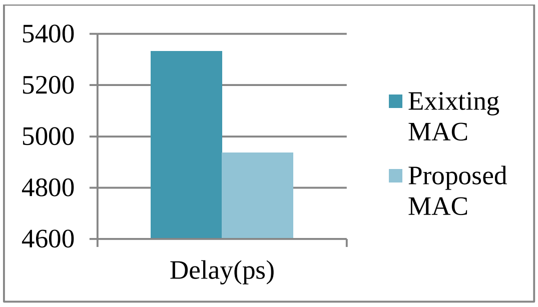

Baseline ▾ = 0
Cursor-Baseline ▾ = 25,000ps

Name	Cursor
a[7:0]	'h AA
b[7:0]	'h 6D
clk	1
mask_x[6:0]	'h 4A
out[16:0]	'h 07FF4
rst	0

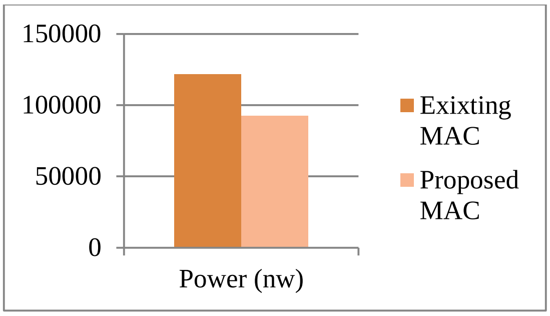
Area comparison between existing and the proposed MAC is shown in below



Delay comparison between existing and the proposed MAC is shown in below



Power comparison between existing and the proposed MAC is shown in below



V. CONCLUSION

In this paper we propose a new approximate multiplier design where it is used in the implementation of MAC unit which can enhance the performance of the MAC. An accuracy-controllable approximate multiplier has been designed on this paper that requires a lot less region and has a reduced path dispose of in evaluation to the conventional layout. Its dynamic accuracy controllability is found out by using manner of the proposed CMA. Both the circuit stage and application diploma are evaluated for the proposed multiplier. The experimental effects display that the proposed multiplier becomes able to supply considerable strength economic savings and speed with the useful resource of maintaining a considerably smaller circuit location than that of the conventional Wallace tree multiplier so that The proposed MAC introduced extra improvements in both power consumption and path delay than other formerly studied approximate MAC units.

REFERENCES

[1] S. Venkataramani, V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan. “Quality programmable vector processors for approximate computing,” 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 1-12, Dec. 2013.

[2] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, “Bio-Inspired imprecise computational blocks for efficient VLSI implementation of Soft-Computing applications,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 4, pp. 850-862, Apr. 2010.

[3] C. Liu, J. Han, and F. Lombardi, “A Low-Power, High-Performance approximate multiplier with configurable partial error recovery,” Design, Automation & Test in Europe Conference & Exhibition (DATE), Mar. 2014.

[4] S. Hashemi, R. I. Bahar, and S. Reda, “DRUM: A Dynamic Range Unbiased Multiplier for approximate applications,” IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 418- 425, Nov. 2015.

[5] B. Moons, M. Verhelst, “DVAS: Dynamic Voltage Accuracy Scaling for increased energy-efficiency in approximate computing,” IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Jul. 2015.

[6] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, “Design and analysis of approximate compressors for multiplication,” IEEE Transactions on Computers, vol. 64, no. 4, pp. 984-994, Apr. 2015.

[7] K. C. Bickerstaff, E. E. Swartzlander, and M. J. Schulte, “Analysis of column compression multipliers,” 15th IEEE Symposium on Computer Arithmetic, pp. 33-39, Jun. 2001.

[8] Z. Yang, J. Han, and F. Lombardi, “Approximate compressors for Error- Resilient multiplier design,” IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), pp. 183-186, Oct. 2015.

[9] NanGate, Inc. NanGate FreePDK45 Open Cell Library, http://www.nangate.com/?page_id=2325, 2008

[10] J. Liang, J. Han, and F. Lombardi, “New metrics for the reliability of approximate and probabilistic adders,” IEEE Transactions on computers, vol. 62, no. 9, pp. 1760-1771, Sep. 2013.

[11] M. S. Lau, K. V. Ling, and Y. C. Chu, “Energy-Aware probabilistic multiplier: Design and Analysis,” 2009 international Conference on Compliers, architecture, and synthesis for embedded systems, pp. 281-290, Oct. 2009.

M.N.Bhavana, B.Tech in ECE from KKC Institute of Technology and Engineering, Puttur in 2017, and PG Scholar (VLSI) in Siddharth Institute of Engineering and Technology, Puttur, AP, India.

J. Prasad Babu, Completed his B.Tech degree from S.V.University, Tirupathi in 2006 and M.tech from PBR Viswodaya Institute of Technology and Science, Kavali in 2012 and Pursuing PhD from Vellore Institute of Technology, AP Amaravathi. He is currently working as an Associate Professor in Department of ECE in Siddharth Institute of Engineering and Technology, Puttur, AP, India.