International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

A Novel Design of 16-Bit MAC Unit using Hybrid Variable Latency CSKA Structure for DSP Applications

P.Samyuktha, Muni Praveena Rela, Dr.M.Gurunadha Babu
[1]M.Tech (VLSI), Department of ECE, CMRIT, Hyderabad, T.S, India
[2]Assoc. professor, Department of ECE, CMRIT, Hyderabad, T.S, India
[3]Professor, Department of ECE, CMRIT, Hyderabad, T.S, India

Abstract- *In this paper, we present an area efficient Multiply and Accumulator (MAC) structure using vedic multiplier and various carry skip adders. In this paper first we examine the area and delays of three types of carry skip adder designs. The first design is conventional CSKA using MUX. Second outline which utilizes applying link and incrementation plans to enhance the proficiency of the traditional CSKA (Conv-CSKA) structure. In addition, rather than using multiplexer logic, the CSKA makes utilization of AND-OR-Invert (AOI) as well as OR-AND-Invert (OAI) compound gates for the skip logic (CI-CSKA). The third CSKA design which implements variable stage size(VSS) CSKA and with the 8-bit parallel prefix adder (PPA) in order to reduce the delay. These CSKA adders are used in MAC implementations along with the 16x16 vedic multiplier and the results are studied. Verilog HDL is used for designing the circuits. The synthesis and simulation results are obtained using Xilinx ISE 14.7.*

Key Words- MAC units, Vedic Multiplier, Carry skip Adders.

I. INTRODUCTION

Due to the rapid growth of portable electronic systems like laptop, calculator, mobile etc, the low power devices have become very important in today's world. Low power and high-throughput hardware configuration are assuming the testing part for VLSI designer. For constant signal processing, a rapid and high throughput MAC unit is dependably a key to accomplish an elite performance digital signal processing framework. The MAC unit performs multiplication and accumulation forms over and again with a specific end goal to perform consistent and complex operations in digital signal processing. MAC unit additionally contains clock and reset so as to control its operation. Various scientists have been concentrating on the plan of advance MAC unit designs. In the adder, the past MAC output and the present output will be included and it comprises of Multiplier unit, one adder unit and both will get to be joined by an aggregate unit. The significant utilizations of Multiply-Accumulate (MAC) unit are processors, logic units and advanced signal processors, since it decides the speed of the general system. The productive plans by MAC unit are Nonlinear Computation like Discrete Cosine or wavelet Transform (DCT), FFT/IFFT. Since, they are fundamentally executed by insistent utilization of multiplication and addition, the whole speed and execution can be processed by the speed of the addition and multiplication occurring in the framework. For the most part the delay, critical delay, occurs because of the long multiplication process and the engendering delay is observed due to parallel adders in the addition organize. High-speed multipliers are also desired as performance of DSP systems is limited by their performance to execute multiplication processes, which is due to the fact that multiplication dominates the execution time of a majority of DSP algorithms [mac3]. Designing of MAC unit that caters to both delay and power issues has, henceforth, become the need of the hour. Vedic Maths was formulated by Swami Bharati Krishna Tirthaji Maharaja from the ancient Indian scriptures (Vedas) after extensive research on the Vedas. Vedic Mathematics finds its base on the sixteen principles or „sutras". Thus, integration of multiplication with Vedic Maths can lead to wonders in various domains of engineering, such as Digital Signal Processing.

II. PRIOR WORK

Since the focus of this paper is on the MAC structure with vedic multiplier and various CSKA designs, first the related work to this MAC are reviewed.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

A. Multiplier Unit

A multiplying function can be carried out in three ways: partial product Generation (PPG), partial product addition(PPA), and final conventional addition. The main bottle neck that should be considered in increasing the speed of MAC are partial product reduction. MAC configuration utilizing traditional multiplier is replaced by vedic multiplier utilizing Urdhava Triyagbhayam sutra. Multiplication is the basic operation of MAC unit [1]. Power utilization, dissipation, area, speed and maintain a strategic distance from them, we go for fast multipliers in different uses of DSP, organizing, and so forth. There is a major rule that enhance the speed of the MAC units which is lessening the partial products and due to that accumulator load is getting decreased. The essential operational area in MAC unit is the multiplier that decides the basic way and the delay. The (log2N + 1) fractional products are created by 2N-1 cross results of various widths for N*N. The fractional products are generated by Urdhava sutra is by Criss Cross Method. The most extreme number of bits in fractional products will prompt Critical way.. The MAC unit with area efficiency at high speed processing with reasonable power consumption, for computation of squares is also studied [2]. This new architecture for MAC unit with low area and high speed was also studied. It was achieved by using Vedic Square [2] existing architecture. The principle of Vedic Square is based on the „Duplex D" property of Urdhva Triyagbhyam which involves addition of twice the product of outermost pair of an n-bit number With odd number of bits, if one bit is left then its square is being taken in the result. Thus, computations decrease with Vedic square reduction in multiplication operations.

B. Addition Block

The existing architecture uses a RCA and the existing architecture, as shown in Fig.1, uses the combination of Vedic multiplier and Sparse Kogge Stone Adder for implementing high speed and low power consumption MAC unit [2].

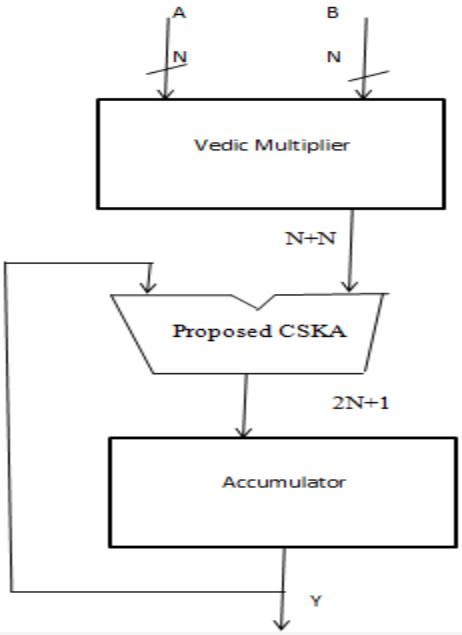


Fig.1: MAC unit

C. Accumulator

Accumulator has a critical part in the DSP applications in different ranges and is an exceptionally essential and common technique. The register planned in the accumulator is utilized to include the multiplied numbers. Multiplier, adder and an accumulator are shaping the basic establishment for the MAC unit. The regular MAC unit has a multiplier and multiplicand to do the essential multiplication and some parallel adders to include the incomplete products produced in the past advance. To get the last multiplication output we add the incomplete product to these outcomes. Vedic Multiplier has put forward to intensify the action of the MAC Unit.

III. PROPOSED MAC STRUCTURE

The design of MAC architecture consists of 3 sub designs.

1. 16×16 bit Vedic multiplier design with proposed adder.
2. Design of 32 bit- CSKA adder using all the 3-different methods.
3. Design of accumulator register which integrates both the results of multiplier and adder stages.

1. Vedic Multiplier

In this section we propose a Vedic multiplication technique called "Urdhva-Tiryakbhyam – Vertically

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

and crosswise." Which can be used not only for decimal multiplication but also used for binary multiplication. This system for the most part comprises of generation of partial products parallel and after that we need to play out the addition operation simultaneously [3]. This algorithm can be used for 2x2, 4x4, 8x8....N×N bit multiplications. Since the wholes and their fractional products are computed in parallel the Vedic multiplier does not relies on the processor clock recurrence. Hence there is no need of increasing the clock frequency and if the clock frequency increases it will automatically leads to the increase in the power dissipation. Consequently by utilizing this Vedic multiplier system we can lessen the power dispersal. The primary favorable position of this Vedic multiplier is that it can lessen delay and in addition zone when contrasted and alternate multipliers. Shift operation is not necessary because the partial product calculation will perform it in a single step, which in turn saves time and power. This is the main advantage of the Vedic multiplier.

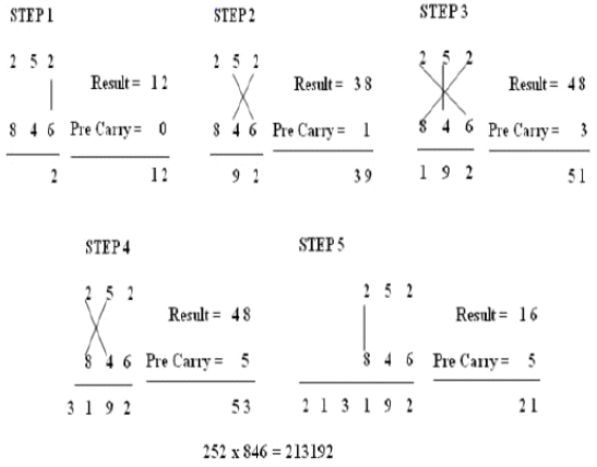


Fig.2 Multiplication of two decimal numbers

A. 2×2 Vedic Multiplier Block

To explain this method let us consider 2 numbers with 2 bits each and the numbers are A and B where A=a0a1 and B=b0b1 as shown in the below line diagram. To begin with the least significant bit (LSB) bit of conclusive product (vertical) is acquired by taking the result of two least significant bit (LSB) bits of An and B is a0b0. Second step is to take the products in a transversely way, for example, the least significant bit (LSB) of the main number A (multiplicand) is increased with the following higher bit of the multiplicand B in an across way. The output produced is 1-Carry bit and 1bit utilized as a part of the outcome as demonstrated as follows. Subsequent step is to take result of 2 most significant bits (MSB) and for the got result already acquired carry to be included. The outcome acquired is utilized as the fourth bit of the last outcome and last carry is the other bit.

$s0=a0b0$ (1)

$c1s1 = a1b0 + a0b1$ (2)

$c2\ s2 = c1 + a1b1$ (3)

The acquired last result is given as c2s2s1s0. A 2×2 Vedic multiplier bit is executed by utilizing two half adders and four two input and gates as appeared in Figure 8.

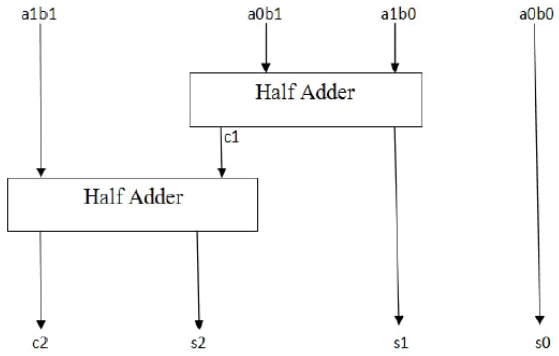


Fig.3 Block Diagram of 2×2 Vedic Multiplier

B. 4x4 Vedic Multiplier Block

In this area, now we will examine around 4x4 bit Vedic multiplier. For clarifying this multiplier let us consider two four bit numbers are An and B to such an extent that the individual bits can be described as the A3A2A1A0 and B3B2B1B0. The strategy for multiplication can be clarified as far as line outline appeared in below figure. The last output can be achieved as the C6S6S5S4S3S2S1S0. The partial products are figured in parallel and thus delay acquired is diminished hugely for the addition in the quantity of bits. The Least Significant Bit (LSB) S0 is found effortlessly by increasing the LSBs of the multiplier and the multiplicand. Here the multiplication is followed according to the steps shown in the line diagram in figure 3. After performing all the steps the result (Sn) and Carry(Cn) is obtained and in the same way at each step the

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

previous stage carry is forwarded to the next stage and the process goes on.

$S0 = A0B0$ (4)

$C1S1 = A1B0 + A0B1$ (5)

$C2S2 = C1 + A0B2 + A2B0 + A1B1$ (6)

$C3S3 = C2 + A0B3 + A3B0 + A1B2 + A2B1$ (7)

$C4S4 = C3 + A1B3 + A3B1 + A2B2$ (8)

$C5S5 = C4 + A3B2 + A2B3$ (9)

$C6S6 = C5 + A3B3$ (10)

For clear understanding, observe the block graphs for 4x4 as appeared in the figure 9 and inside the block chart 4x4 absolutely there are four 2x2 Vedic multiplier modules, and three ripple carry adders which are of four bit estimate are utilized. The four bit ripple carry adders are utilized for addition of two four bits and in like manner absolutely four are use at adjacent stages 3 of multiplier. The carry created from the primary ripple carry adder is passed on to the following ripple carry adder and there are two zero contributions for second ripple carry adder. The course of action of these adders are appeared in block graph which can diminishes the computational time with the end goal that the delay can be diminish.

Hence this is the general numerical solution appropriate to all instances of multiplication and its equipment design is appeared in fig. 5. So as to multiply two 8-bit numbers utilizing 4-bit multiplier we continue as takes after. Consider two 8 bit numbers denoted as AHAL and BHBL where AH and BH corresponds to the most significant 4 bits, AL and BL are the least significant 4 bits of an 8-bit number.

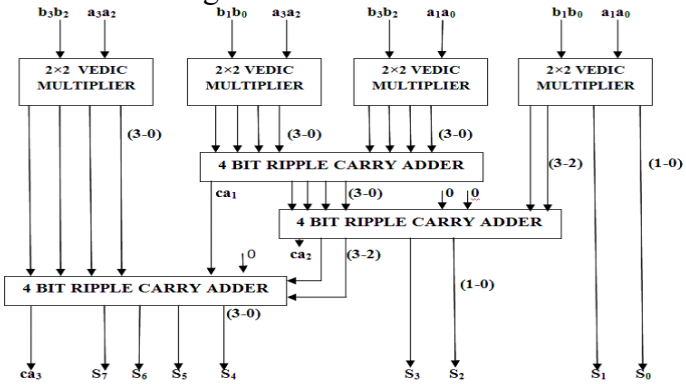


Fig.4 Block Diagram of 4x4 bit Vedic Multiplier

At the point when the numbers are multiplied by UrdhavaTiryakbhyam (vertically and crosswire) strategy, we get, AH AL BH BL (AH x BH) + (AH x BL + BH x AL) + (AL x BL). Hence we require four 4-bit multipliers and two adders to include the partial products and 4-bit intermediate carry created. Since result of a 4 x 4 multiplier is 8 bits in length, in each progression the least significant 4 bits compare to the product and the rest of the 4 bits are given to the following stage. This process continues for 3 steps in this case. Similarly, 16 bit multiplier has four 8 x 8 multiplier and three 16 bit adders with 8 bit carry. Consequently we see that the multiplier is profoundly measured in nature. Henceforth it prompts consistency and versatility of the multiplier design. The accompanying fig.5 demonstrates the design of a 16×16 Vedic multiplier utilizing a 8×8 Vedic multiplier and the outline can be actualized utilizing Verilog HDL.

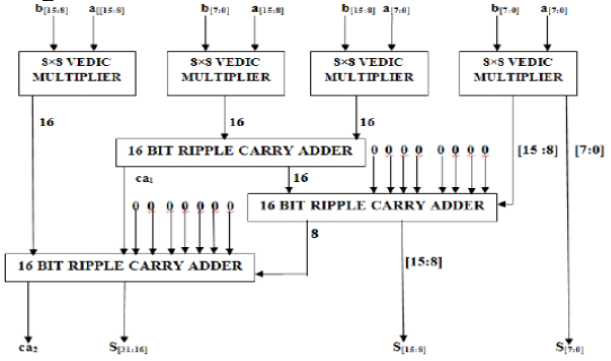


Fig.5 Block diagram of 16x16 Vedic Multiplier

2. Design Of Various Carry Skip Adders

1) Designing a CSKA structure using the conventional CSKA (Conv-CSKA) structure.

2) Designing a modified CSKA structure by combining the concatenation and the incrementation schemes to the conventional CSKA (Conv-CSKA). structure for improving the speed and energy proficiency of the adder. The change furnishes us with the capacity to utilize easier carry skip logics in light of the AOI/OAI compound entryways rather than the multiplexer. 3) Designing a half and half factor variable CSKA structure in light of the addition of the recommended CSKA, by adding instead a portion of the center stages in its structure with a PPA, this is altered in [4].

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

A. Modifying CSKAs for Improving Speed

The ordinary structure of the CSKA comprises of stages containing chain of full adders (FAs) (RCA block) and 2:1 multiplexer. The RCA blocks are associated with each other through 2:1 multiplexers, which can be put into at least one level structures [5]. The CSKA setup (i.e., the quantity of the FAs per organize) greatly affects the speed of this kind of adder. Numerous techniques have been recommended for finding the ideal number of the FAs. The procedures exhibited in [5] make utilization of VSSs to limit the postponement of adders in light of a solitary level carry skip logics. These techniques, however, cause area and power increase considerably and less regular layout.

Alioto and Palumbo [5] propose a straightforward system for the design of a solitary level CSKA. The strategy depends on the VSS system where the close ideal quantities of the FAs are resolved in view of the skip time, and the ripple time (the time required by a bring to ripple through a FA). The objective of this strategy is to diminish the basic way delay by considering a noninteger proportion of the skip time to the ripple time on in opposition to a large portion of the past work. In all of the works reviewed so far, the focus was on the speed, while the power consumption and area usage of the CSKAs were not considered. Also for the speed, the delay of skip logic, which depend on multiplexers and frame a vast part of the adder basic way delay [5], has not been decreased.

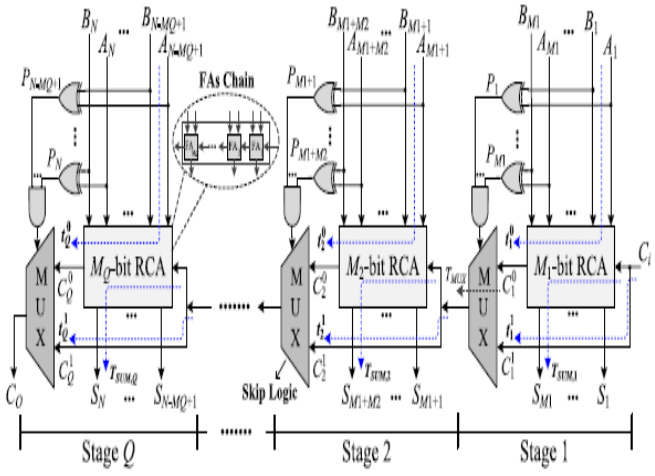


Fig.6 Conventional Carry Skip Adder[4]

B. Structure of modified CI-CSKA

The structure depends on joining the link and the incrementation designs [6] with the ConvCSKA structure, and thus, is meant by CI-CSKA. It gives us the capacity to utilize more straightforward carry skip logics. The logic replaces 2:1 multiplexers by AOI/OAI compound gates (Fig. 7). The gates, which comprise of less transistors, have bring down delay, zone, and littler power utilization contrasted and those of the 2:1 multiplexer [7]. Note that, in this structure, as the carry spreads through the skip logics, it progresses toward becoming supplemented. Therefore, at the output of the skip logic of even stages, the complement of the carry is generated. The structure has a significant lower engendering delay with a marginally littler region contrasted and those of the conventional one. Note that while the power utilizations of the AOI (or OAI) gate are littler than that of the multiplexer, the power utilization of the proposed CI-CSKA is somewhat more than that of the ordinary one. This is because of the addition in the quantity of the gates, which forces a higher wiring capacitance (in the noncritical ways).

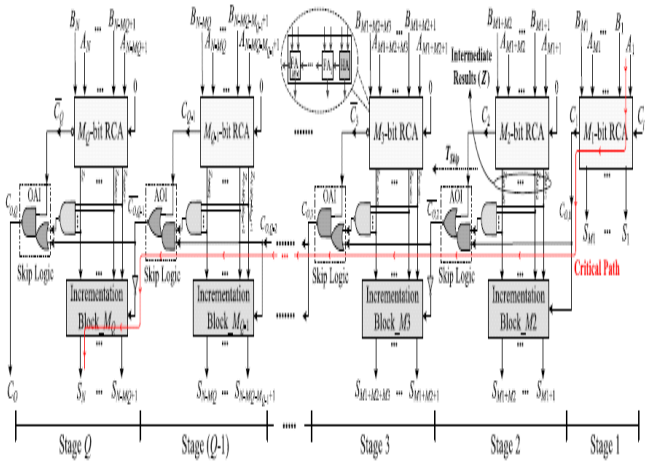


Fig.7 Modified CI-CSKA Structure

The explanation behind utilizing both AOI and OAI compound gates as the skip logics is the modifying elements of these gates in standard cell libraries. Thus the requirement for an inverter, which expands the power utilization and delay, is wiped out. As appeared in Fig.5, if an AOI is utilized as the skip logic, the following skip logic should utilize OAI

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

gate. Furthermore, another point to say is that the utilization of the proposed skipping structure in the Conv-CSKA structure builds the delay of the basic way extensively. This begins from the way that, in the Conv-CSKA, the skip logic can't sidestep the zero carry contribution until the point when the zero carry input spreads from the comparing RCA block. To solve this issue, in the proposed structure, we have utilized a RCA block with a carry contribution of zero (utilizing the link approach). Along these lines, since the RCA block of the stage does not have to sit tight for the carry output of the past stage, the output brings of the blocks are computed in parallel.

C. Hybrid Variable Latency CSKA Structure

The essential thought behind utilizing VSS CSKA structures depended on nearly adjusting the delays of ways to such an extent that the delay of the basic way is limited contrasted and that of the FSS structure [8]. This denies us from having the chance of utilizing the slack time for the supply voltage scaling. To give the variable inactivity highlight to the VSS CSKA structure, we supplant a portion of the center stages in our proposed structure with a PPA altered in this paper. It has to be noticed that since the Conv-CSKA structure has a lower speed than that of the proposed one, in this area, we don't think about the ordinary structure. The proposed hybrid variable latency CSKA structure is appeared in Fig. 8 where a Mp-bit changed PPA is utilized for the pth arrange. Since the core arrange, which has the biggest size among the stages, is available in both SLP1 and SLP2, supplanting it by the PPA diminishes the delay of the longest off-basic ways. Subsequently, the utilization of the quick PPA enables expanding the accessible slack time in the variable latency structure. It should be included that since the input bits of the PPA block are utilized as a part of the indicator block, these block moves toward becoming parts of both SLP1 and SLP2.

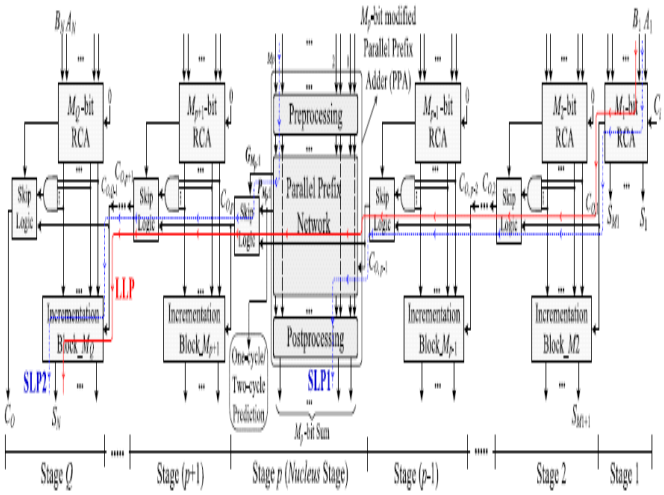
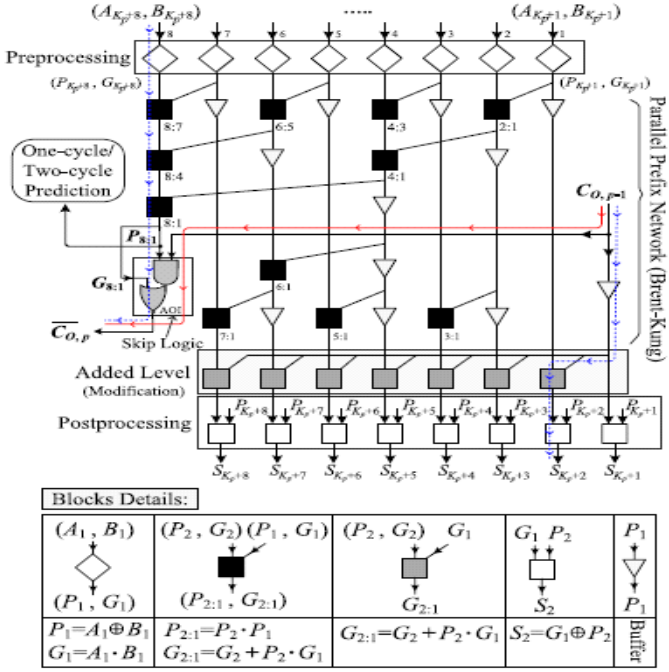


Fig.8 Hybrid variable size CI-CSKA

In the hybrid structure, the prefix network of the Brent–Kung adder is used for constructing the nucleus stage (Fig. 8). One the benefits of this adder contrasted and other prefix adders is that in this structure, utilizing forward paths, the longest carry is ascertained sooner contrasted and the intermediate passes, which are processed by in reverse paths. In addition, the fan-out of adder is lesser than other parallel adders, while the length of its wiring is smaller [2]. Finally, it has a simple and regular layout.



International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

Fig.9 Internal structure of the pth stage of the proposed hybrid variable latency CSKA

As appeared in the figure, in the preprocessing level, the propagate signals (Pi) and generate signals (Gi) for the information sources are computed. In the following level, utilizing Brent– Kung parallel prefix arrange, the longest carry (i.e., G8:1) of the prefix organize alongside P8:1, which is the result of the all engender signs of the input sources, are computed sooner than other middle signals in this system. The signal P8:1 is utilized as a part of the skip logic to decide whether the carry output of the past stage (i.e., CO,p-1) ought to be skipped or not. Furthermore, this signal is abused as the indicator signal in the variable latency adder.

It should be represented that these operations are performed in parallel with different stages. For the situation, where P8:1 is one, CO,p-1 should avoid this stage foreseeing that some basic paths are started processing. Then again, when P8:1 is zero, CO,p is equivalent to the G8:1. Likewise, no basic path will be initiated for this situation. After the parallel prefix organize, the middle carries, which are elements of CO,p-1 and moderate signals, are processed (Fig. 9). At long last, in the post processing level, the output aggregates of this stage are figured. It should be noticed that this usage depends on the comparative thoughts of the connection and incrementation ideas utilized as a part of the CI-CSKA discussed. It should be noticed that the end some portion of the SPL1 path from CO,p-1 to conclusive summation aftereffects of the PPA block and the starting block of the SPL2 paths from contributions of this part to CO,p have a place with the PPA block (Fig. 9). And, like the proposed CI-CSKA structure, the main purpose of SPL1 is the primary input bit of the principal arrange, and the last purpose of SPL2 is the last bit of the whole output of the incrementation block of the stage Q. Since the PPA structure is more efficient when its size is equal to an integer power of two, we can select a larger size for the nucleus stage accordingly [2]. The bigger size (number of bits), contrasted and that of the core arrange in the first CI-CSKA structure, prompts the decline in the quantity of stages also littler delays for SLP1 and SLP2.

3. Accumulator

The accumulator is designed to store cumulative addition of MAC unit, It is a group of registers which are designed for this. It has a reset pin which is used for resetting. When the reset value is high the content of the accumulator becomes zero and when reset is not equal to zero, the accumulator starts accumulating the summation. The inputs to the accumulator are output from the vedic multiplier and the previous content of the accumulator. Schematics and

IV. SIMULATIONS RESULTS

In this section, first we will see the RTL Schematics of the designed MAC unit using vedic multiplier and various CSKA implementation and their simulation results. Then we can compare the delays and area utilized by the proposed structures. The vedic multiplier considered here are the 16x16-bit vedic multiplier and the adder designed here are 32-bit adders. These were designed using Verilog HDL and simulated using Xilinx ISE 14.4. The RTL schematics of the MAC unit and all the sub modules like vedic multiplier and all the CSKA structures are given below.

RTL Schematics

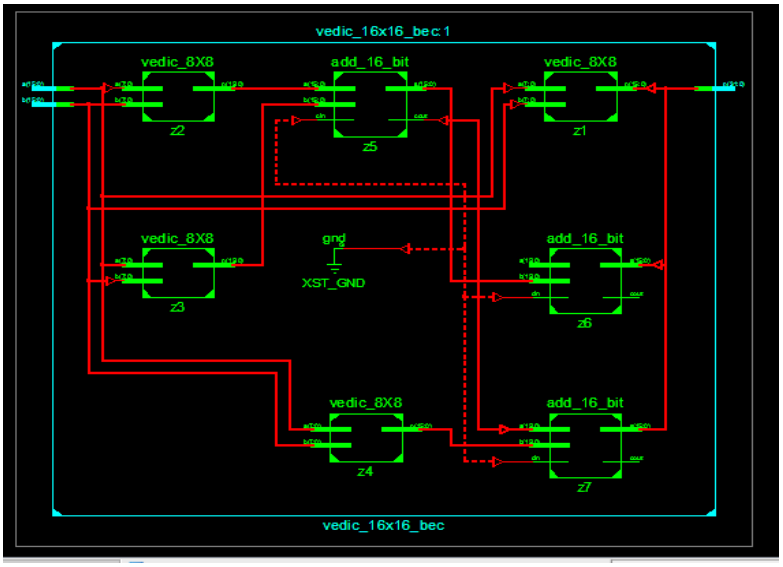


Fig.10 RTL Schematic of Vedic Multiplier

The above fig.10 represents RTL schematic of 16 bit Vedic Multiplier with sub instances 4-8x8 Vedic Multipliers and 3-proposed 16 bit adders and from these instances we will get 32 bit product output from the Vedic Multiplier.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

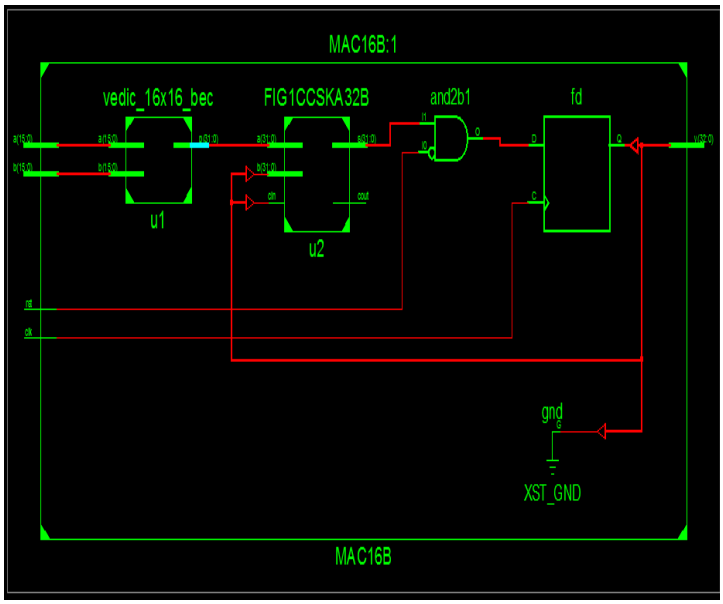


Fig.11 RTL Schematic of MAC with conv-CSKA

The above fig.11 represents RTL schematic of 16 bit MAC unit using sub instances 1-16x16 Vedic Multipliers and 32- conventional-CSKA and from these instances we will get 32 bit accumulated output from the top MAC unit design.

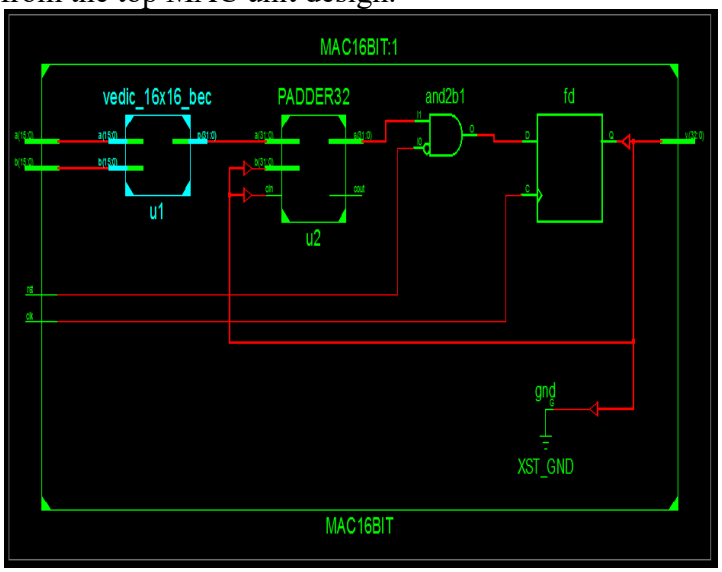


Fig.12 RTL Schematic of MAC with CI-CSKA

The above fig.12 represents RTL schematic of 16 bit MAC unit using sub instances 1-16x16 Vedic Multipliers and 32- Proposed-CSKA and from these instances we will get 32 bit accumulated output from the top MAC unit design.

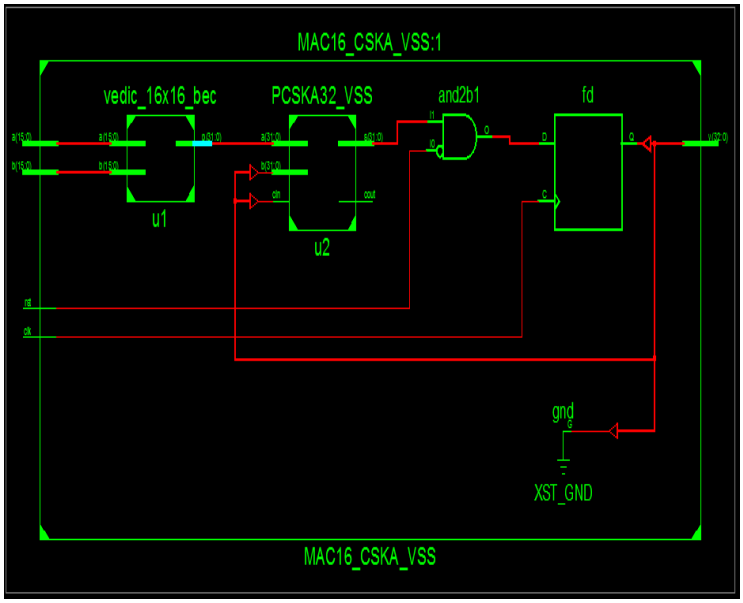


Fig.13 RTL Schematic of MAC with hybrid CSKA

The above fig.13 represents RTL schematic of 16 bit MAC unit using sub instances 1-16x16 Vedic Multipliers and proposed 32- Hybrid-CSKA and from these instances we will get 32 bit accumulated output from the top MAC unit design.

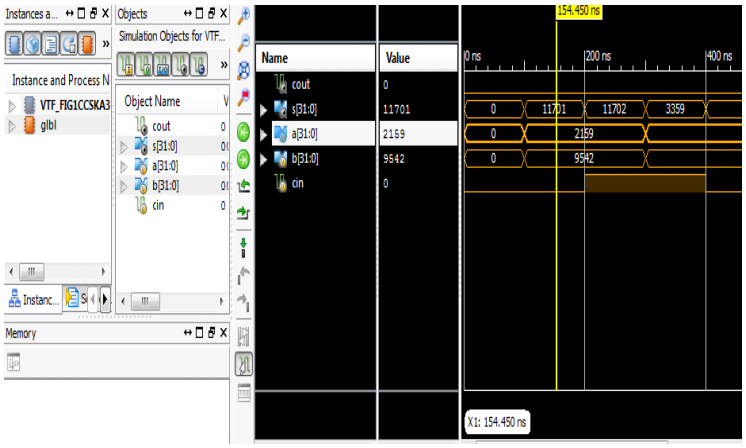


Fig.14 Simulation of Conv-CSKA

In Fig.14 a=2159, b=9542 and cin=0 then sum=11701, if cin=1 then sum value will be 11702.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

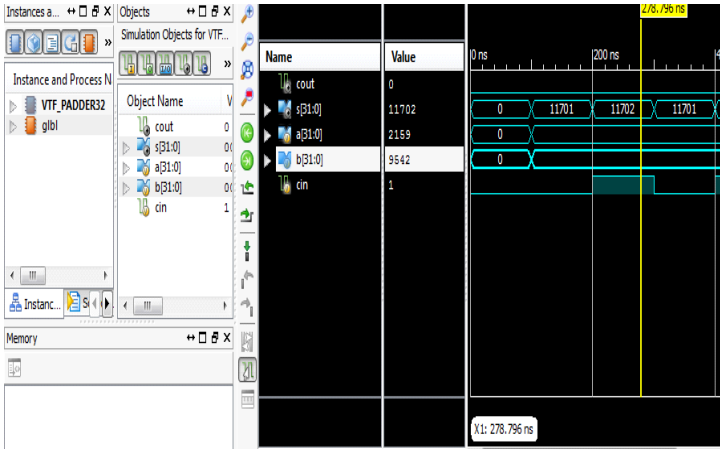


Fig.15 Simulation of CI-CSKA

In Fig.15 a=2159, b=9542 and cin=0 then we will get sum=11701, if cin=1 then we will get sum value will be added with 1 then sum=11702.

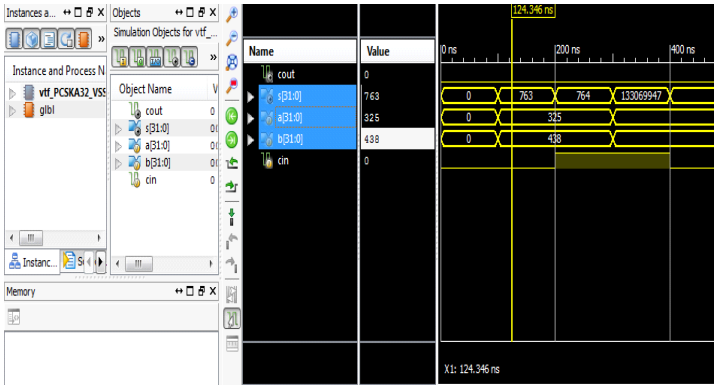


Fig.16 Simulation of CSKA

In Fig.16 a=325 and b=438 and cin=0 then sum=763, if cin=1 then sum value will be 764.

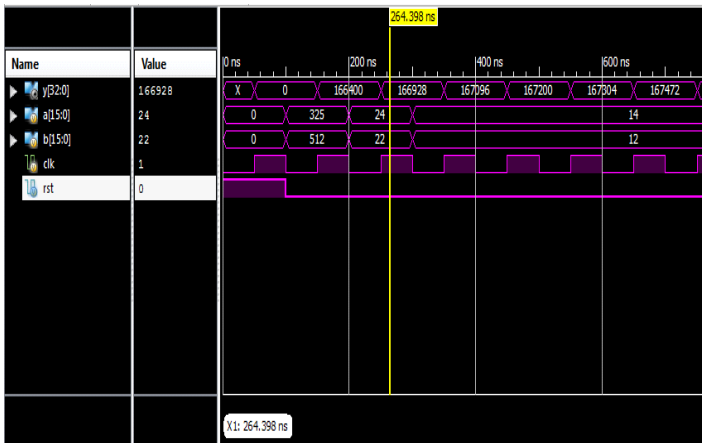


Fig.17 Simulation of MAC unit hybrid CSKA

In Fig.17 a=325 and b=512 and rst=1 then we will get product value is 166400. The another value of a=24 and b=24 then the product value is 528, 528 and previous accumulation value 166400 is added and the final value of accumulation register for the next clock is 166928 like that accumulation register send previous value as a one input for the adder. And adder adds present product value and previous accumulation value then send input for the accumulation register for next iteration.

Delay and Area Comparisons of CSKAs The delays of various CSKAs are compared as shown in below Table-I. It is noted that the conv- CSKA occupies less no LUTs but delay is more. In the CI-CSKA structure occupies slightly higher number of LUTs but the delay is considerably less. In the hybrid CSKA structure the delay is reduced and numbers of LUTs are also reduced.

Table-I Delay and Area Comparisons results of 3-CSK Adders.

Adder Design	No of slices	No of LUT's	Delay(ns)
Conventional [19]	51	88	54.94
Proposed [FSS]	57	101	34.67
Proposed [VSS]	55	98	21.80

V. CONCLUSION

The MAC unit designed with vedic multiplier and different carry skip adders are analyzed and compared for parameters like Area, and Delay Consumption. In terms of delay, MAC with convCSKA and hybrid CSKA have almost same delay. But if you consider the area perspective the hybrid CSKA occupies less number of LUTs.

REFERENCES

[1] AvisekSen, ParthaMitra, DebarshiDatta, "Low Power MAC Unit for DSP Processor", International Journal of Recent Technology and Engineering (IJRTE) Vol. 1, Issue-6, January 2013, pp. 93-95.

[2] Gitika Bhatia, Karanbir Singh Bhatia, OsheenChauhan, SoumyaChourasia, Pradeep

Kuma5,"A efficient MAC unit with low area consumption" IEEE INDICON ,2015.

[3] BhavaniPrasad.Y, Ganesh Chokkakula, SrikanthReddy.P and Samhitha.N.R, Design of Low Power and High Speed Modified Carry Select Adder for 16 bit Vedic Multiplier, ISBN No.978-1-4799-3834-6/14/$31.00©2014.

[4] Samir Palinitkar,"Verilog HDL: A Guide to Digital Design and Synthesis".

[5] M. Alioto and G. Palumbo, "A simple strategy for optimized design of one-level carry-skip adders," IEEE Trans. Circuits Syst. I, Fundam.Theory Appl., vol. 50, no. 1, pp. 141–148, Jan. 2003.

[6] MarojuSaiKumar, D.Ashok Kumar, Dr.P.Samundiswary,"Design and performance of analysis of Multiply and Accumulate Unit",ICCPCT 2014.

[7] Milad Bahadori, Mehdi Kamal, Ali AfzaliKusha, and MassoudPedram,"Hi-speed and energy efficient carry skip adders operating under a wide range of supply voltage levels", IEEE Trans,2015.

[8] Premananda B.S. , Samarth S. Pai, Shashank B, Shashank S. Bhat, "Design and Implementation of 8-Bit Vedic Multiplier" International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 12, December 2013.