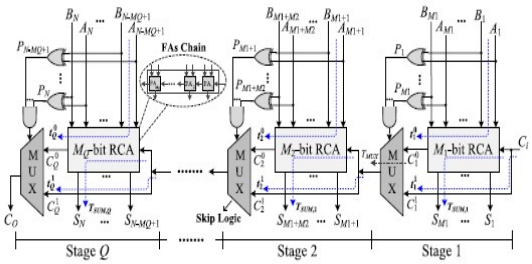

Fig.4 Conventional Carry Skip Adder[4]

- **Structure of modified CI-CSKA**

The structure is based on combining the concatenation and the incrementation schemes [13] with the Conv-CSKA structure, and hence, is denoted by CI-CSKA. It provides us with the ability to use simpler carry skip logics. The logic replaces 2:1 multiplexers by AOI/OAI compound gates (Fig. 5). The gates, which consist of fewer transistors, have lower delay, area, and smaller power consumption compared with those of the 2:1 multiplexer. Note that, in this structure, as the carry propagates through the skip logics, it becomes complemented. Therefore, at the output of the skip logic of even stages, the complement of the carry is generated. The structure has a considerable lower propagation delay with a slightly smaller area compared with those of the conventional one. Note that while the power consumptions of the AOI (or OAI) gate are smaller than that of the multiplexer, the power consumption of the proposed CI-CSKA is a little more than that of the conventional one. This is due to the increase in the number of the gates, which imposes a higher wiring capacitance (in the noncritical paths).

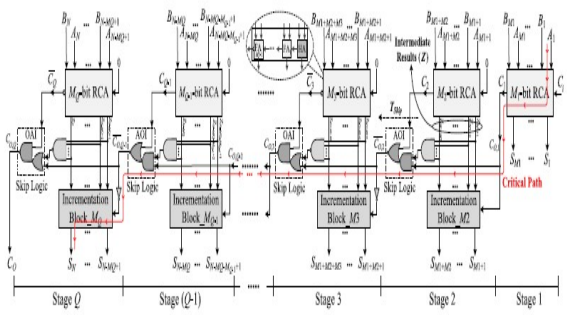

Fig. 5 Modified CI-CSKA Structure

The reason for using both AOI and OAI compound gates as the skip logics is the inverting functions of these gates in standard cell libraries. This way the need for an inverter gate, which increases the power consumption and delay, is eliminated. As shown in Fig. 5, if an AOI is used as the skip logic, the next skip logic should use OAI gate. In addition, another point to mention is that the use of the proposed skipping structure in the Conv-CSKA structure increases the delay of the critical path considerably. This originates from the fact that, in the Conv-CSKA, the skip logic (AOI or OAI compound gates) is not able to bypass the zero carry input until the zero carry input propagates from the corresponding RCA block. To solve this problem, in the proposed structure, we have used an RCA block with a carry input of zero (using the concatenation approach). This way, since the RCA block of the stage does not need to wait for the carry output of the previous stage, the output carries of the blocks are calculated in parallel.

- ***H*ybrid Variable Latency CSKA Structure**

The basic idea behind using VSS CSKA structures was based on almost balancing the delays of paths such that the delay of the critical path is minimized compared with that of the FSS structure. This deprives us from having the opportunity of using the slack time for the supply voltage scaling. To provide the variable latency feature for the VSS CSKA structure, we replace some of the middle stages in our proposed structure with a PPA modified in this paper. It should be noted that since the Conv-CSKA structure has a lower speed than that of the proposed one, in this section, we do not consider the conventional structure. The proposed hybrid variable latency CSKA structure is shown in Fig. 6 where an *Mp*-bit modified PPA is used for the *p*th stage (nucleus stage). Since the nucleus stage, which has the largest size (and delay) among the stages, is present in both SLP1 and SLP2, replacing it by the PPA reduces the delay of the longest off-critical paths. Thus, the use of the fast PPA helps increasing the available slack time in the variable latency structure. It should be mentioned that since the input bits of the PPA block are used in the predictor block, this block becomes parts of both SLP1 and SLP2.

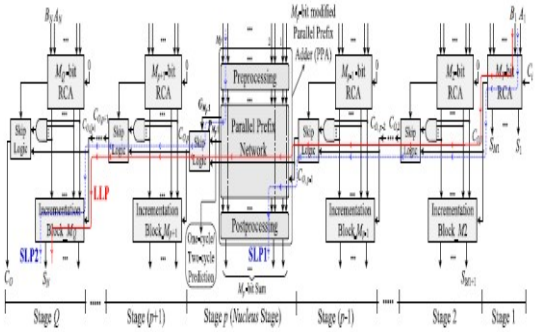

Fig. 6. Hybrid variable size CI-CSKA

In the hybrid structure, the prefix network of the Brent–Kung adder is used for constructing the nucleus stage (Fig. 6). One the advantages of the this adder compared with other prefix adders is that in this structure, using forward paths, the longest carry is calculated sooner compared with the intermediate carries, which are computed by backward paths. In addition, the fan-out of adder is lesser than other parallel adders, while the length of its wiring is smaller [14]. Finally, it has a simple and regular layout.

In the preprocessing level, the propagate signals (*Pi)* and generate signals (*Gi)* for the inputs are calculated. In the next level, using Brent–Kung parallel prefix network, the longest carry (i.e., *G*8:1*)* of the prefix network along with *P*8:1, which is the product of the all propagate signals of the

inputs, are calculated sooner than other intermediate signals in this network. The signal $P8{:}1$ is used in the skip logic to determine if the carry output of the previous stage (i.e., $CO,p-1$) should be skipped or not. In addition, this signal is exploited as the predictor signal in the variable latency adder. It should be mentioned that all of these operations are performed in parallel with other stages. In the case, where $P8{:}1$ is one, $CO,p-1$ should skip this stage predicting that some critical paths are activated. On the other hand, when $P8{:}1$ is zero, CO,p is equal to the $G8{:}1$. In addition, no critical path will be activated in this case. After the parallel prefix network, the intermediate carries, which are functions of $CO,p-1$ and intermediate signals, are computed (Fig. 7). Finally, in the postprocessing level, the output sums of this stage are calculated. It should be noted that this implementation is based on the similar ideas of the concatenation and incrementation concepts used in the CI- CSKA discussed. It should be noted that the end part of the SPL1 path from $CO,p-1$ to final summation results of the PPA block and the beginning part of the SPL2 paths from inputs of this block to CO, p belong to the PPA block (Fig. 7). In addition, similar to the proposed CI-CSKA structure, the first point of SPL1 is the first input bit of the first stage, and the last point of SPL2 is the last bit of the sum 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 [14]. The larger size (number of bits), compared with that of the nucleus stage in the original CI-CSKA structure, leads to the decrease in the number of stages as well smaller delays for SLP1 and SLP2.

- **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.

IV. Schematics and 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**

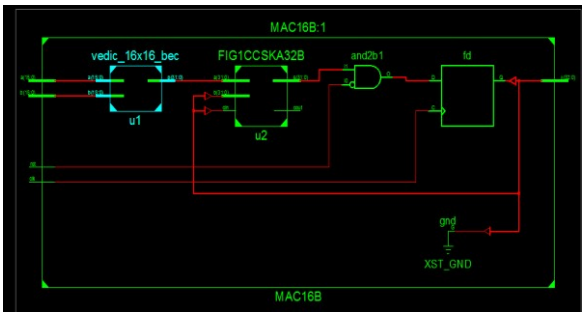

Fig. 9 . RTL Schematic of MAC with conv-CSKA

- **Simulation Results of Proposed MAC units**

Fig, 10 Simulation Result of MAC unit with Conv- CSKA

- **Delay and Area Comparisons of CSKAs**

The delays of various CSKAs are compared below.

Adder Design	No. of Slices	No. of LUTs	Delay(ns)
32 bit Conv-CSKSA	51	88	70.94
32 bit CI-CSKA	57	107	46.67
32 bit Hybrid CSKA	55	98	27.80

Fig.11 Comparison of Area and Delay of CSKAs

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 number of LUTs are also reduced.

- **Delay *Comparisons MACs with different CSKAs***

The delays of MAC units implemented with different CSKAs are compared below.

MAC Design	Logic Delay(ns)	Route Delay(ns)	Total Delay(ns)
MAC with Con-CSKA	20.669	37.891	58.560
MAC with CI-CSKA	21.412	40.525	61.936
MAC with Hybrid CSKA	21.02	37.630	58.659

Fig. 12 Delay Comparison of MAC units

It is noted that the total delay of MAC with conv-CSKA and MAC with hybrid CSKA are almost same. In the case of MAC with CI-CSKA structure delay is more.

- **Area Comparison of MACs with different CSKAs**

The Area calculations of MAC units implemented with different CSKAs are compared below. It is noted that the area occupied by MAC with conv-CSKA is more compared to other two structures.

MAC Design	No. of Slices	No. of LUTs
MAC with Conv-CSKSA	494	884
MAC with CI-CSKA	487	867
MAC with Hybrid CSKA	477	851

Fig. 13 Comparison of Area of MAC units.

Name of the Gate	MAC with Conv-CSKA	MAC with CI-CSKA	MAC with Hybrid CSKA
AND Gate	1434	1420	1415
OR Gate	65	61	61
XOR Gate	92	92	87
NAND Gate	6	8	6
NOR Gate	3	4	3
NOT Gate	3	4	8

Fig. 14 . Comparison of Area of MAC Unit in terms of Gate Count.

The number of gates used for implementing MAC units with different CSKAs are compared above.

- It is noted that the number of gates used by MAC with conv- CSKA is more compared to other two structures. MAC unit with hybrid CSKA occupies less number of slices and LUTs.

V. CONCLUSIONS

The three MAC unit are designed with vedic multiplier and three carry skip adder designs, They are analyzed and compared for parameters like Area and Delay. When you consider the area perspective, the MAC unit designed with hybrid CSKA occupies less number of LUTs among the three MAC designs. In terms of delay, MAC with conv-CSKA and hybrid CSKA have almost same delay. When u consider only the CSKA designs the hybrid CSKA has very less delay while conventional CSKA has more delay. But when you implement those CSKA designs in a MAC unit, there is no much difference in the delay because the multiplier always decides the critical path and determines the speed of the overall hardware systems.

References

[1] N. H. E. Weste and D. Harris, CMOS VLSI Design, 4th edition, Pearson–Addison-Wesley, 2011

[2] Koren, Computer Arithmetic Algorithms, 2nd ed. Natick, MA, USA: A K Peters, Ltd., 2002.

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

[4] Milad Bahadori, Mehdi Kamal, Ali Afzali-Kusha, and Massoud Pedram,"Hi-speed and energy efficient carry skip adders operating under a wide range of supply voltage levels", IEEE Trans,2015.

[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] David H.K.Hoe, Chris Martinez and Sri Jyothsna Vundavalli", Design and Characterization of Parallel Prefix Adders using FPGAs", 2011 IEEE 43rd South eastern Symposium in pp. 168-172, 2011

[7] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," IEEE Trans.Comput., vol. C-22, no. 8, pp. 786–793, Aug. 1973

[8] V. G. Oklobdzija, B. R. Zeydel, H. Dao, S. Mathew, and R. Krishnamurthy, "Energy-delay estimation technique for high performance microprocessor VLSI adders," in Proc. 16th IEEE Symp.Comput. Arithmetic, Jun. 2003, pp. 272–279.

[9] Sudheer Kumar Yezerla, B Rajendra Naik,Design and Estimation of Delay, Area and Power for Parallel Prefix Adders, Proceedings of 2014 RAECS UIET Punjab University Chandigarh, 06 - 08 March, 2014

[10] Bhavani Prasad.Y, Ganesh Chokkakula, Srikanth Reddy.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.

[11] 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

[12] Avisek Sen, Partha Mitra, Debarshi Datta, "Low Power MAC Unit for DSP Processor", International Journal of Recent Technology and Engineering (IJRTE) Vol. 1, Issue-6, January 2013, pp. 93-95

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

[14] Gitika Bhatia, Karanbir Singh Bhatia, Osheen Chauhan, Soumya Chourasia, Pradeep Kumar,"A efficient MAC unit with low area consumption" IEEE INDICON ,2015.

M.Ganesh Kumar , I.V.Rameswar Reddy, K.Kameswara Reddy," Design of Two Variable Multiplier Using Vedic Mathematics and ROM Approach", IJERT 2013