



High Speed Adder-Multiplier Unit with S-MB Recoding

Riboy Cherian¹, Amel Siby²

Associate Professor, Dept. of ECE, Saintgits College of Engineering, Kottayam, Kerala, India¹

PG Student [VLSI], Dept. of ECE, Saintgits College of Engineering, Kottayam, Kerala, India²

ABSTRACT: Adders and multipliers are the fundamental units used in signal processing. Hence the allocation and its architecture largely affect the performance of the signal processing units. The adders and multipliers are separately optimized to improve the performance of the signal processing units. In most of the signal processing units adders are followed by multipliers, and hence the adder-multiplier units are formed. A technique called S-MB recoding which convert the inputs to be added directly to modified booth form without adding them is discussed in the paper. An adder-multiplier unit to enhance its speed has been proposed. Also the existing adder-multiplier unit is compared with proposed unit in terms of speed. Both the units have been modelled in VHDL, simulated and synthesized in Xilinx ISE 14.5.

KEYWORDS: Carry Lookahead Adder, Carry Save Adder, Carry Select Adder, Kogge stone based carry select adder, Wallace carry save adder.

I.INTRODUCTION

As the technology is being advancing faster there is a fast advancement in multimedia and communication systems that demands for advancement in real time signal processing. It is a technology that enables to hold the fundamental theory, applications, algorithms, and implementations of processing or transferring information contained in many different physical, symbolic, or abstract formats broadly designated as signals. Since there is a need for advancement in signal processing especially digital signal processing that demand for large capacity data processing, many research has been going on to make these processing the most efficient as possible. The signals processing functions are realised by the repetitive use of addition and multiplication. The performance of the signal processing unit is largely affected by the decision regarding the allocation and architecture of the addition and multiplication unit Kostas Tsoumanis et. al, 2014. In most of the above digital signal processing functions addition is often followed by multiplication, hence to achieve good performance and execution speed adder-multiplier (AMU) was introduced. By doing so, compared to conventional ones more efficient implementation of digital signal processing was achieved. Fastest adders with low area, high speed and power requirements have been used. To speed up the multiplication operation many recording schemes are used to reduce the partial products and sum up or to accumulate the partial products. Partial product generation can be done in two ways. The first approach is the direct generation in which two input and gate is used for generating the partial product directly. The second one is based on the booth algorithm. After the partial product has been generated, next step is the partial product reduction which can be achieved by using partial product reduction tree. After the partial product reduction step, two rows remain for accumulation which can be achieved by using a carry propagate adder (CPA). By optimizing the three parts of the multiplier better performance can be achieved.

In this paper an adder-multiplier unit to optimize its speed is proposed. The adder and Modified Booth encoding unit are fused into a single unit to form the S-MB recoding unit. S-MB recoding is done to convert the sum to booth recoding. Booth recoding is done to reduce the number of partial products. The partial products are accumulated by using the carry select adder and the final sum and carry is accumulated by using a carry lookahead adder. To improve the speed of the adder-multiplier unit, in this paper the accumulation of the final sum and carry is done by a kogge stone based carry select adder. Both designs are modeled in VHDL, simulated and synthesized in Xilinx ISE 14.5. The synthesis result is used to study the performance of Adder-Multiplier unit in terms of area and delay.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2015

II.LITERATURE SURVEY

Several techniques has been used to optimize the performance of the multiply-add unit in terms of area, power consumption and delay. In a paper by Y-H. Chen et.al 2010 design of a parallel MAC unit based on radix-2 modified booth algorithm is designed. Here multiplication is combined with accumulation a hybrid type of carry save adder is used to improve the performance. The carry save adder tree uses a one's compliment based radix-2 modified booth algorithm for partial product generation and accumulation. With the intension to increase the bit density of operands a modified array for sign extension is used. In order to reduce the number of bits of final adder, the carry save adder is responsible for propagating the carry to the least significant bit in advance.

In another paper by Li-Hsun Chen et. al 2005 a multiply-add unit that uses radix-4 booth recording is used. Optimized compressors are used for carry save addition in order to avoid the use of half adders to reduce the hardware complexity. A circuit to detect the dynamic range of the inputs to the multiplier is designed and the one with small dynamic range is used for the booth recording. This is done to increase the probability of the partial product to be zero to reduce the switching activity of the multiply-add unit. Also the effective dynamic range of output from multiplier is compared with the input to be added and larger dynamic range is selected as effective word length of for addition. It is mainly focused to reduce the power consumption. The direct implementation of multiply-add unit by first adding the input and multiplying the result with other input to the multiplier increases both the area and delay of the unit. To avoid this fusion technique which is based on direct recording of the result of addition is used.

One of the most advanced type of MAC unit has been proposed by F. Elgui in which a dependence graph is used to represent the multiply-add unit and the partial product generation is done by using modified booth algorithm. Here accumulation is combined with a carry save adder tree to compress the partial products. In this paper the number of input bits to the final adder is reduced and the adder for accumulation is eliminated so that the critical path can be reduced. But the problem with this structure is that since final adder results are used for accumulation the output rate need to be improved.

C. N Lyu et. al 1995 proposed a redundant binary booth recording technique in which redundant binary inputs are transformed to their corresponding modified booth encoding form. With regard to keeping the operands in carry save representation a special expansion of pre-processing step of recorder is required. R. Zimmerman et. al 2003 proposed an optimized design for the one discussed before which helps to improve both area and critical path.

In another paper proposed by Ayman A. Fayed et. al 2002 a high speed multiply-add unit has been designed. Here a [4:2] compressor cell is used to improve the speed. The [4:2] compressor cells are not fully utilized due to the parallelogram shape of the partial products as they do not get the complete four input. Here the unused inputs of the [4:2] compressor cell is fed with the accumulated data so that addition can be merged with the multiplication. Hence the speed can be increased, area can be saved and power consumption is reduced.

III.PROPOSED ADDER-MULTIPLIER UNIT

The inputs which are to be added is coded to booth form with the help of S-MB (Sum to Modified Booth) recoding. With booth recording, the number of partial products will be reduced. With the modified booth bits and the input X the partial products are generated and they are accumulated by using a carry save adder tree. The proposed method optimizes the adder-multiplier unit in terms of speed by using a kogge stone based carry select adder.

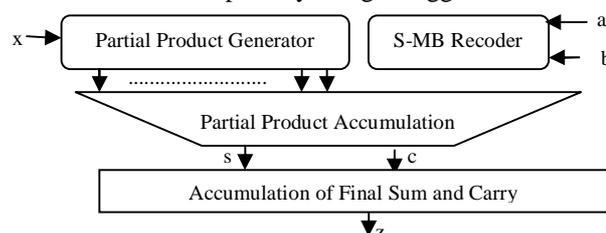


Fig 1: Adder-Multiplier Unit

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2015

3.1 S-MB RECODING

In S –MB recoding technique, recoding is done on the sum of two consecutive bits of inputs A(a_{2j}, a_{2j+1}) with the two consecutive bits of input B(b_{2j}, b_{2j+1}) into single modified booth digit Y_j^{MB} said Kostas Tsoumanis et. al, 2014 in their paper. In modified booth recoding three bits are needed to form the modified booth digit and also the most significant bit among them is negatively weighted and the other two bits are having positive weights. Hence a signed bit arithmetic is needed in order to transform the above mentioned pairs of bits to modified booth form. Hence by considering the inputs to be signed a bit level signed half adder and full adder is being developed

This recoding scheme uses special type of half adder HA^* and full adder FA^* HA^* adder accepts two binary inputs P,Q and produces the binary output S,C. Here in HA^* adder the sum is considered negative and the output can take values $\{0, +1, +2\}$. It is used to implement the relation $2.C-S = P+Q$.

$$\text{Sum, } S = P \text{ xor } Q \tag{3.1}$$

$$\text{Carry, } C = P \text{ xor } Q \tag{3.2}$$

FA^* adder accepts three binary inputs P, Q, Ci and produces the binary output S,C. Here in FA^* adder the sum and input bit Q is considered as negative and the output can take values $\{-1, 0, +1, +2\}$. It is used to implement the relation $2.C-S = P - Q + Ci$. Conventional FAs are used to implement the signed FAs with inputs being negative and inverting the outputs.

$$\text{Sum, } S = P \text{ xor } Q \text{ xor } Ci \tag{3.3}$$

$$\text{Carry, } C = ((P \text{ or } (\text{not } Q)) \text{ and } Ci) \text{ or } (P \text{ and } (\text{not } Q)) \tag{3.4}$$

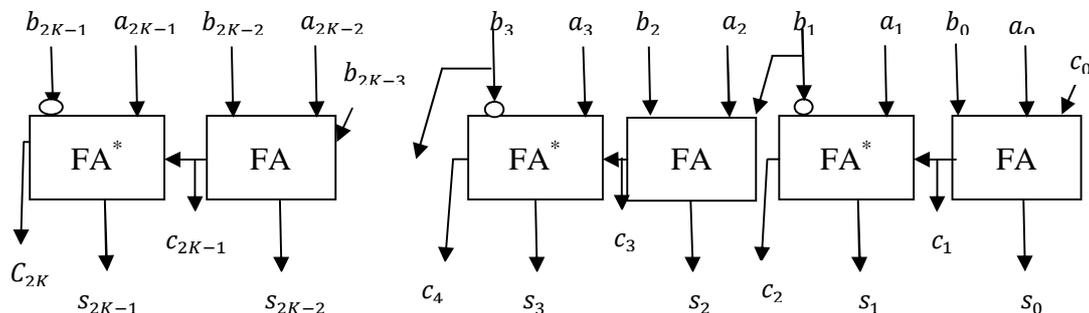


Fig 2: S-MB Recoding Scheme

The figure 2 shows the S-MB recoding scheme. To determine the modified booth digit the S-MB cell is cascaded as shown fig 3.5. In order to determine the least significant modified booth digit, the c_0 input of the conventional adder of the first S-MB cell is assigned to be '0'. Other inputs are a_0 and b_0 which produce the sum bit s_0 and carry bit c_1 . This carry together with inputs a_1 and b_1 are given to the signed full adder to produce the output sum bit s_1 and carry bit c_2 .

The least significant modified booth bit is determined by

$$Y_0^{MB} = -2 s_1 + s_0 + 0 \tag{3.5}$$

Hence in general terms, the inputs b_{2j-1}, b_{2j}, a_{2j} bits are given as input to the conventional full adder to generate the

$$\text{Sum bit, } S_{2j} = a_{2j} \text{ xor } b_{2j} \text{ xor } b_{2j-1} \tag{3.6}$$

$$\text{Carry bit, } c_{2j+1} = (a_{2j} \text{ and } b_{2j}) \text{ or } (b_{2j-1} \text{ and } (a_{2j} \text{ or } b_{2j})) \tag{3.7}$$

This carry bit c_{2j+1} together with bits b_{2j+1}, a_{2j+1} is given as input to the signed full adder to produce the output

$$\text{Sum bit, } S_{2j+1} = a_{2j+1} \text{ xor } b_{2j+1} \text{ xor } c_{2j+1} \tag{3.8}$$

$$\text{Carry bit, } c_{2j+2} = (a_{2j+1} \text{ and } (\text{not } b_{2j+1})) \text{ or } (c_{2j+1} \text{ and } (a_{2j+1} \text{ or } (\text{not } b_{2j+1}))) \tag{3.9}$$

Since the bit s_{2j+1} is negatively weighted we use the signed full adder FA^* . Generally the modified booth digit is determined by

$$Y_j^{MB} = -2 s_{2j+1} + s_{2j} + c_{2j} \tag{3.10}$$

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2015

The most significant modified booth digit is determined by the equation

$$Y_k^{MSD} = -a_{2k-1} + c_{2k} \tag{3.11}$$

3.2 PARTIAL PRODUCT GENERATOR

S-MB recoder converts the inputs A & B of 2K bits to be added and whose result to be multiplied with X directly into modified booth digit without adding the inputs A & B to reduce the number of partial products. Now the modified booth digit is multiplied with the input X for producing the partial product. The partial products are generated based on the modified booth bit.

Table 3.4: Partial product generation table

MODIFIED BOOTH BIT	PARTIAL PRODUCT
0	0*X
+1	1*X
-1	2's compliment(1*X)
+2	left shift (1*x)
-2	2's compliment(left shift(1*X))

After the partial products are generated, they are appended with zero or one towards their left side depending on the value of s_j ie.sign extension is done.

3.3 PARTIAL PRODUCT ACCUMULATION

A carry save adder is used to accumulate the partial products resulting in the final sum and carry. It will improve the speed of accumulation of the partial product since it saves the carry and give it to the next level of carry select adder. Hence the adders in the same layer becomes independent of each other and can be executed simultaneously. Hence the time required for the addition operation is reduced. To improve the speed of the accumulation process a Wallace carry save adder is used. In Wallace carry select adder first three partial products are given to the first carry select adder. The next three partial products are given to the next carry select adder and so on. After accumulation of partial products the result will be a final sum and carry.

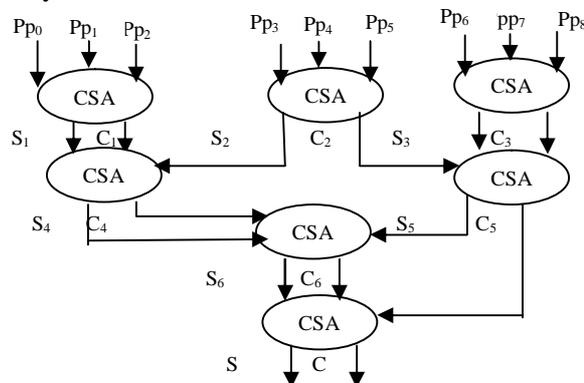


Fig 3: Wallace Carry Save Adder

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2015

3.4 ACCUMULATION OF FINAL SUM AND CARRY

With the intension to improve the speed of the adder-multiplier unit the kogge stone based carry select adder is used to accumulate the final sum and carry. Kogge stone based carry select adder is a fast adder designed by the combination of carry select adder and kogge stone adder. Kogge stone adder is the fastest adder design possible. But it has the disadvantage that it occupies large area. Carry select adder is an adder which is a compromise for both area as well as speed. Hence by combining the two adders high speed can be achieved along with a compromise for the large area occupancy of the kogge stone adder. Carry select adder is an adder that provides a compromise the small area occupancy as well as large delay of the carry propagate adder and the small delay with large area occupancy of the carry lookahead adder. The main components of the carry select adder is a full adder and a multiplexer

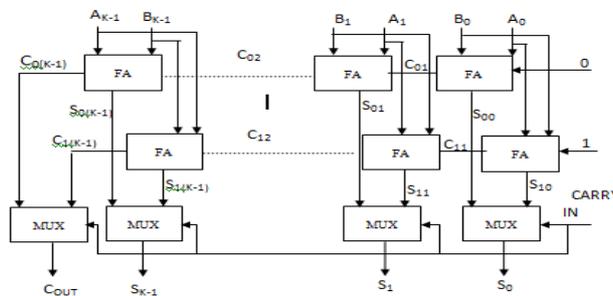


Fig 4: K-Bit Carry Select Adder

Kogge stone adder is the fastest adder design possible. Kogge stone adder is a parallel prefix adder. They differ from carry lookahead adder in the carry generation stage. Here processing is done in a parallel fashion, hence fastest computation is possible. It has minimum logical depth as well as fan-out since carries are generated in parallel. The kogge stone adder mainly have the generate and propagate block, Black cell Block and Gray cell block. The working of kogge stone adder is classified into three parts.

(i) Pre-Processing:-

Corresponding to each pair of input bits generate and propagate signals are calculated. Consider that A and B are the inputs which are to be added is K bit wide. The logic behind the computation of generate and propagate signal is as follows:-

$$\text{Generate signal, } g_i = A_i \text{ and } B_i \quad (3.12)$$

$$\text{Propagate Signal, } p_i = A_i \text{ xor } B_i \quad (3.13)$$

(ii) Carry Lookahead Network:-

Here computation of carries corresponding to each bit takes place. This is the step that differentiates the kogge stone adder from other adders. This step is responsible for the high performance of the kogge stone adder. Here computation of group generate and propagate signal is done as intermediate signals. It is calculated as follows

$$P_{i:j} = P_{i:K} \text{ and } P_{K:j} \quad (3.14)$$

$$G_{i:j} = G_{i:K+1} \text{ or } (P_{i:K+1} \text{ and } G_{K:j}) \quad (3.15)$$

(iii) Post Processing:-

Here the sum bit is calculated and is the final step. The sum bit is given as below

$$s_i = p_i \text{ xor } c_{i-1} \quad (3.16)$$

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2015

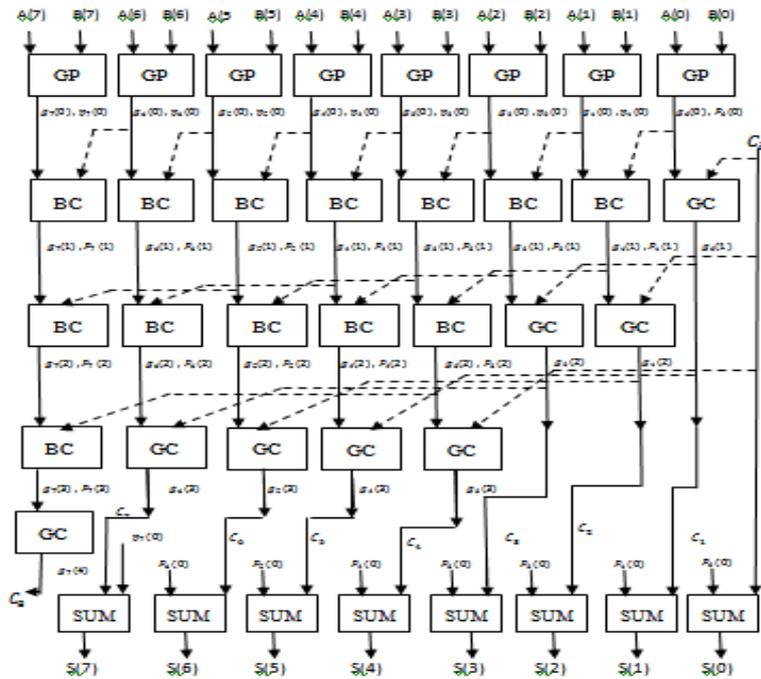


Fig 5: Kogge stone adder

In the proposed adder-multiplier unit a kogge stone based carry select adder is used in order to improve the speed. In kogge stone based carry select adder the full adders used to compute the sum in carry select adder is replaced by kogge stone adders. In doing so the speed is increased as well as large area occupied by the kogge stone adder is compromised. The advantage of using the kogge stone based carry select adder is that the number of processing steps are reduced and hence the delay will be reduced to produce the result of final sum and carry.

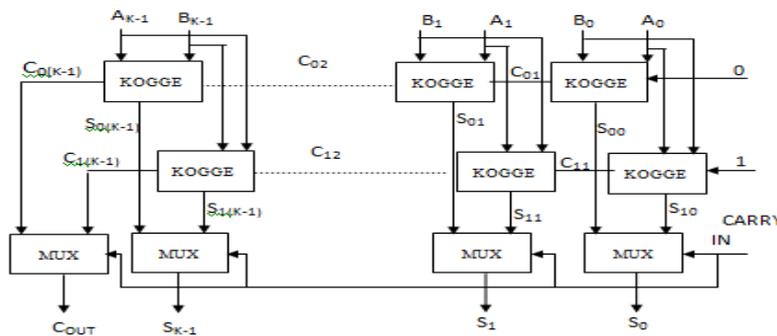


Fig 6: Kogge stone based carry select adder

III. RESULT AND DISCUSSION

Adder-multiplier unit with CLA and kogge stone based carry select adder has been designed in VHDL, simulated and also synthesized by using Xilinx ISE 14.5. These are designed for 16 and 32 bit and their total combinational path delay has been compared. Adder-multiplier unit with CLA and kogge stone based carry select adder is designed first for 16 and 32 bit inputs. After that the next step is the synthesis of the design that has been modeled with the help of the test bench written and the output waveforms are obtained to test the correctness of the design. Now the synthesis of the design is being done and from the synthesis result the maximum combinational logic delay is known and the two architectures are compared.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2015

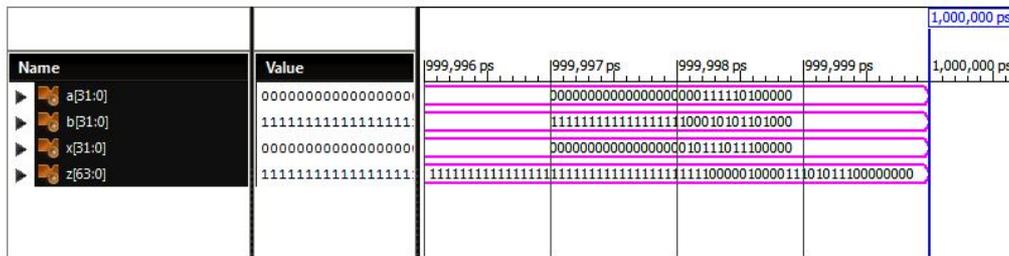


Fig 7: Simulation Waveform

Fig.7 gives the simulation waveform of the Adder-Multiplier unit. Simulation is done with a=4000, b= -15000 and x=12000 for both 16 bit and 32 bit inputs.

Table 2: Synthesis results of adder-multiplier unit with CLA and kogge stone based carry select adder

No. of bits	Architecture	Maximum Combinational Path Delay (ns)
16-bit	Adder-multiplier with CLA	16.347
	Adder-multiplier with kogge stone based CSA	12.261
32-bit	Adder-multiplier with CLA	33.373
	Adder-multiplier with kogge stone based CSA	15.784

Table 2 gives the maximum combinational path delays of the adder-multiplier unit with CLA and kogge stone based carry select adder for both 16 bit and 32 bit inputs. The synthesis result shows that adder-multiplier unit with kogge stone based carry select adder has more speed compared to the one with carry lookahead adder. The results shows that the difference in the maximum combinational path delay between adder-multiplier unit with carry lookahead adder and kogge stone based carry select adder is more while moving to higher number of input bits. And the kogge stone based carry select adder is having more speed.

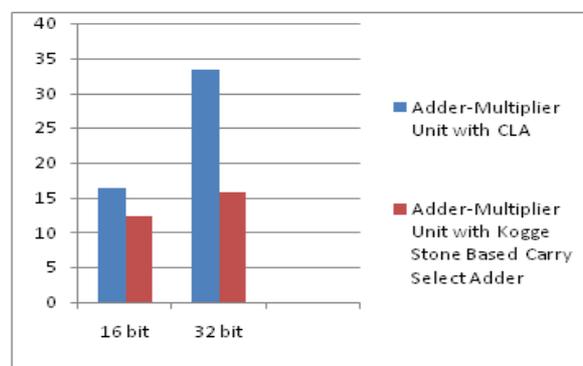


Fig. 8: Comparison of Delay Between Adder-Multiplier unit with CLA and Kogge stone based carry select adder

Fig. 8 gives the chart representation of the table 2. From the graph it is clearly evident that the speed of the adder-multiplier unit is improved by using kogge stone based carry select adder compared to carry lookahead adder. As the number of bits increases the speed improvement also increases.



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2015

IV.CONCLUSION

An adder-multiplier unit with S-MB recoding is being modeled. S-MB recoding techniques converts the inputs to be added directly to modified booth form without adding them. The result is then used to form the partial product which are then accumulated by using carry save adders. The resulting sum and carry are then accumulated by using a kogge stone based carry select adder instead of carry lookahead adder with the intension to improve the speed. Both the adder multiplier unit with CLA and kogge stone based carry select adder has been modeled for both 16 bit and 32 bit inputs. The design for both the adder-multiplier unit was done in VHDL using Xilinx ISE 14.5. Simulation results shows that adder-multiplier unit with kogge stone based carry select adder is more faster than the one with CLA. This speed improvement increases as the number of bits increases.

REFERENCES

- [1] Begum, Mohammed Haseena, and V. Vamsi Mohana Krishna. "Design And Verification Of Low Power And Area Efficient Kogge-Stone Carry Select Adder." In International Journal of Engineering Research and Technology, ERSRA Publications, vol. 2, no. 8, pp. 462-467, Aug 2013.
- [2] Chakali, Pakkiraiah, and Madhu Kumar Patnala. "Design of High Speed Kogge-Stone Based Carry Select Adder." International Journal of Emerging Science and Engineering (IJESE), vol. 1, no. 4, pp. 34-37, Feb. 2013.
- [3] Chen, OT-C., RR-B. Sheen, and Sandy Wang. "A low-power adder operating on effective dynamic data ranges." Very Large Scale Integration (VLSI) Systems, IEEE Transactions, vol. 10, no. 4, pp. 435-453, 2002.
- [4] CH. Sudharani and CH. Ramesh. "Design And Implementation Of High Performance Parallel Prefix Adders." In International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, no. 9, Sept. 2014.
- [5] Elguibaly, Fayed. "A fast parallel multiplier-accumulator using the modified Booth algorithm." Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions, vol. 47, no. 9, pp. 902-908, Sept. 2000.
- [6] Fayed, Ayman A., and Magdy A. Bayoumi. "A merged multiplier-accumulator for high speed signal processing applications." In Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference , vol. 3, pp. 3212-3215, 2002.
- [7] J Seo, Young-Ho, and Dong-Wook Kim. "A new VLSI architecture of parallel multiplier-accumulator based on Radix-2 modified Booth algorithm." Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 18, no. 2, pp. 201-208, 2010.
- [8] JUNETA, PUSHKAL. "Parallel Implementation of a 4 X 4-bit Multiplier Using Modified Booth's Algorithm." IEEE Journal of solid-state circuits, vol. 23, no. 4, pp. 1010-1013, Aug.1988.
- [9] Mala, T. Ratna, R. Vinay Kumar, and T. Chandra Kala. "Design and Verification of Area Efficient High-Speed Carry Select Adder." IJRCCCT, vol. 1, no. 6, pp. 345-349, Nov. 2012.
- [10] Sunil M, Ankit R D, Manjunatha G D and Premananda B S " Design And Implementation of Fast Parallel Prefix Kogge Stone Adder." In International Journal of Electrical and Electronics Engineering & Telecommunications, vol. 3, no. 1, January 2014.
- [11] Wallace, Christopher S. "A suggestion for a fast multiplier." Electronic Computers, IEEE Transactions, pp. 14-17, 1964.
- [12] Zicari, Paolo, Stefania Perri, P. Corsonello, and G. Cocorullo. "An optimized adder accumulator for high speed MACs." In ASIC, 2005. ASICON 2005, 6th International Conference vol. 2, pp. 703-706, 2005.
- [13] Zimmermann, Reto, and David Q. Tran. "Optimized synthesis of sum-of-products." In Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference , vol. 1, pp. 867-872, 2003.
- [14] Parhami, Behrooz. Computer arithmetic. Oxford university press, pp. 93-97, 1999.
- [15] N. H. Weste, D. Harris, A. Banarjee "Cmos VLSI Design", Pearson Education, pp. 436-438, 480-486, 2006.