Design of High Speed Approximate Multiplierusing Adder Compressors

T.Vishnu Murthy

Assistant professor, Department of Electronics and communication Engineering,

Pragathi Engineering college, Surampalem, Andhra Pradesh, India

T.Sri.Veerendra

P G scholar , Department of Electronics and communication Engineering

Pragathi Engineering college, Surampalem, Andhra Pradesh, India

Abstract- In this modern era, many of the digital systems are error resilient which allows us to take the advantage of approximate computations. This makes the use of replacement of exact computing units by their counterparts. Approximate computing can also decrease the complexity at the designing levels with an increase in performance and power efficiency. Adders and multipliers are the basic buildings blocks of many digital applications. These blocks can be approximated in several ways. Research works are on the rise at many levels on approximate computing. Approximation at designing level is more advantageous as the modifications at this level much easier than the preceding levels. A method of designing an approximate multiplier with a novel structure introduced in 16-bit adder compressor is proposed. The 16-bit adder compressor (AC) is designed with 8-2 adder compressors in general. The 8-2 adder compressor is designed with 7-2 and 3-2 adder compressors and half adders. The existing and proposed multiplier is designed using Xilinx 14.7 in the frontend. The speed of proposed multiplier 55.44% increase compare to Existing Multiplier.

Keywords: ***7*-2 and 3-2 compressor, low power multiplier, approximation.**

1. INTRODUCTION

Need for approximation arises from the fact that exact computation requires more energy. That means wherever the accuracy is not a major concern and the design has to be energy efficient, we may take the advantage of approximation which requires less energy compared to the exact one. For most of the digital circuits adders and multipliers are the basic building blocks. Replacing the exact building blocks with approximate ones results in energy-efficient designs. A multiplier is a device which multiplies any two operands and gives the corresponding result. Multiplication is nothing but the repeated addition of partial products. This involves the addition of partial products by the use of half adders and full adders based on the bit-size of input operands. Logic gates are used to implement these adder circuits under different technologies. In the design of high-speed multipliers, compressors are used in the reduction tree to speed up the process. Basically, these compressors are implemented using full adders. Moreover, Integrated circuit(IC) era of emerging digital trends prefers compact size. This ensures the necessity of area efficient designs for most of the digital circuits. At the same time, it allows the approximate values but not the exact one for the implementation of energy efficient designs. To make the most of error tolerance, various techniques are available. These are of three types: (1) insistent voltage scaling; (2) truncation of bit-width; (3) use of imprecise building blocks.

The concept in [1] is the use of imperfect full adder cells to implement the multi-bit adder cells with minimized complexity at transistor level. Reflection of errors due to approximation in a typical digital processing at higher levels may not impact the output quality much. Two new methods for approximate 4-2 compressor are proposed in [2] for implementing a multiplier and are analyzed for a dadda multiplier. Simulation of these methods at 1GHz frequency revealed that these methods reduce power, delay and transistor count significantly. It uses XOR-XNOR combinations for the implementation of the compressor. [3] proposed approximate multipliers for DSP applications. This technique takes „m‟ concomitant bits (i.e., m-bit segment) from each n-bit operand where m is greater than or equal to „n/2‟. An m-bit segment can start only from one of two or three fixed bit positions depending on where the leading one bit is located for a positive number. This can provide much higher accuracy. Approximate multiplier circuits proposed in [4] use the technique of partial product perforation. In this technique the errors are bound and predictable. This approach can be used for any multiplier regardless of its architecture. Perforation skips the generation of partial products instead of cutting them. Thus decrease the number of operands to be accumulated and reduces delay. An approximate multiplier with configurable partial error recovery is proposed in [5]. This mainly focuses on the mitigation of critical path by using only simple but fast adders in the reduction tree. An inaccurate 4-2 counter is used for error correction in the implementation of multiplier proposed in [6].The existing multiplier[7] proposed two multipliers, one is for approximating all columns, and second one approximates the least significant columns. It is based on the probability statistics of the logic block results. This paper presents a new approach for approximate multiplier design using 16-bit AC.

2. EXISTING WORK

Existing method evolves the partial products to instigate the terms with different probabilities. These probabilities are analysed and then approximated in a logical method. Based on probability statistics, actual partial products are accumulated to get generate signals using OR gates. These generate signals have severe impact on error probability. To achieve the exact results, the maximum number of „generate' signals that can be grouped is limited to „4'.Remaining signals other than the „g' signals are accumulated by using the approximate 4-2 compressor, half adder, and full adders. To reduce the area and delay, the existing method used an OR gate in the place of X-OR for sum in the half adder. This results in one error in sum computation. In case of full adder, X-OR gates are replaced with OR gates in sum calculation. In 4-2 compressor for every three X-OR gates one X-or is replaced with OR gate. This results in 5 wrong cases out of each 16 cases. With these blocks, two multipliers are designed. In first multiplier; the approximation is done in all columns of partial products of n-bit multiplier. The second multiplier approximation is applied in n-1 least significant OR gates thus reduces the area and power columns. Both the multipliers are designed simple significantly but failed to give accurate values for higher product values.

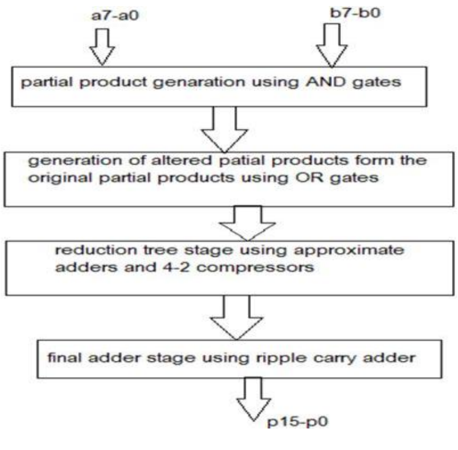

a7-a0
b7-b0
partial product genaration using AND gates
generation of altered patial products form the original partial products using OR gates
reduction tree stage using approximate adders and 4-2 compressors
final adder stage using ripple carry adder
p15-p0

Fig. 1 Block diagram of the existing multiplier

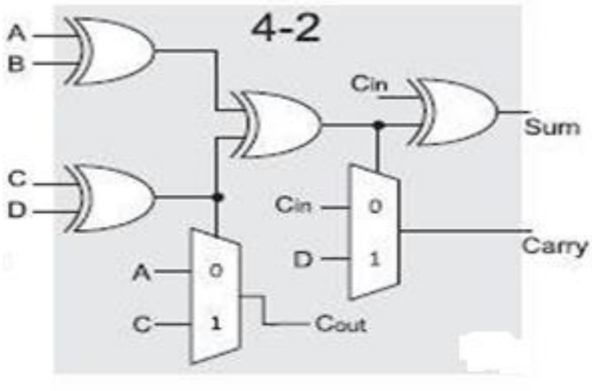


Fig.2: Existing multipliers 4-2compressor block

To overcome this, 4-2 compressor module in the existing method and improves speed is slightly modified in the proposed method. Also the full adder used in the design of compressor is implemented using two techniques. The proposed Multiplier is designed in Xilinx on front-end. Implementation of the multiplier is carried out in the following steps: obtaining partial products, converting partial products into propagate and generate signals, combining the obtained „p" and „g" signals by suitable logic blocks, i.e., half adder, full adder, 4-2 compressor .Consider two 8-bit unsigned operands $\alpha=\sum^{7}_{m=0}\alpha_m 2^m$; and $\beta=\sum^{7}_{n=0}\beta_n 2^n$. Then the partial products are obtained by performing AND operation between α and β. i.e., $a_{m,\ n}=\alpha_m.\beta_n$. Resultant partial products are as shown in below Fig. 2.

3. EXISTING APPROXIMATE MULTIPLIER

The partial products $a_{m,n}$ and $a_{n,m}$ in the columns containing more than three partial products are combined to get propagate and generate signals as shown. These form the altered partial products $p_{m,n}$ and $g_{m,n.}$ These are obtained as follows:

$$P_{m,n} = a_{m,n} + a_{n,m}$$

$$G_{m,n} = a_{m,n}.\ a_{n,m}$$

The reduction tree for obtaining partial products is shown in below Fig.3:

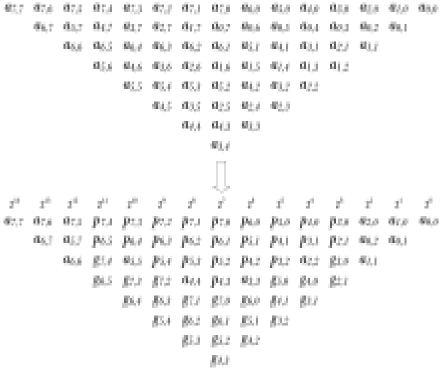


Fig.3. altered partial products formed by propagate and generate signals. [7]

In the process of approximating the altered partial products, "generate" signals are accumulated column-wise using OR gates. OR gates used for a column having mgenerate signals are m/4.Partial products other than the generate signals are approximated using half adder, full adder and 4-2 compressor. In the Existing method approximate half adder and full adder blocks are designed using Adaptive Voltage Level technique which is used to reduce the power consumption. In general, the 4-2 compressor is designed using two full adders. To improve the accuracy, the proposed method includes a half adder with an XOR gate at the output. The Existing Method partial product reduction stage diagram see in Fig. 4.

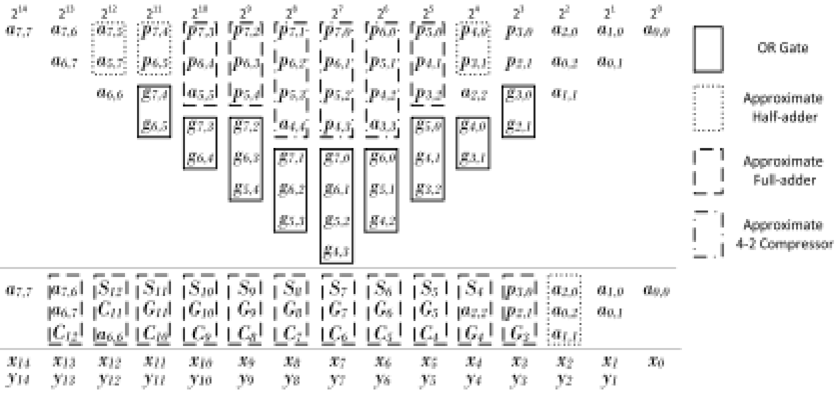


Fig. 4: Logic blocks used for approximation of altered partial products [7]

4 ADDER COMPRESSORS

Compressors by far have been considered as the most efficient building blocks of a high speed multiplier. It provides an advantage of accumulation of partial products at an expense of least possible power dissipation. Rather than entirely summoning partial products with the help of CSA/Ripple adder tree, a structure of compressors would complete the same task in much lesser time and also will simultaneously eradicate the problems of large power consumption and optimization of the area. This addition of partial products when done using conventional method of implementing Full Adders and Half Adders cannot account as much to lessening of delay associated with the critical path as when counter or compressors are used. The reason for the apparent preference of compressors over counters is the advantages it provides in terms of power, number of transistors used and the delay associated with the critical path (comprising of XOR's mainly). The compressor design implemented in this paper prefers both MUX’s and XOR’s.

The internal structure of the 3-2 adder compressor is presented in Fig. 5-a. The maximum delay is given by two XOR gates. The final sum S of the 3-2 adder compressor is given in (1). The 3-2 adder compressor can also be used as a full-adder (i.e. mux-based full-adder) when the input C is used as a carry input.

$$S=Sum+2Carry \tag{1}$$

The internal structure of the 7-2 adder compressor is presented in Fig. 5-b. The maximum delay is given by ten XOR gates. The final sum S of the 7-2 adder compressor is given in (2).

$$S=Sum+2(Cout1+Cout2+Carry) \qquad (2)$$

In this paper 8-2 adder design using 3-2 and 7-2. The internal structure of the 8-2 adder compressor is presented in Fig.6 . The final sum S of the 8-2 adder compressor is given in (3)

.

$$S=Sum+2(Cout0+Cout1+Cout2+Cout3+Cout4+Carry) \qquad (3)$$

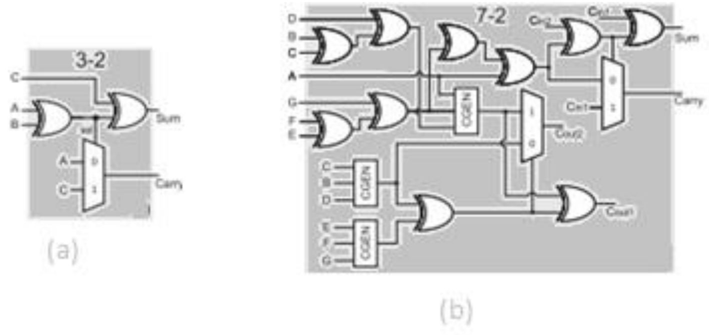


Fig. 5 Adder compressors internal structures: (a) 3-2; (b) 7-2. [8]

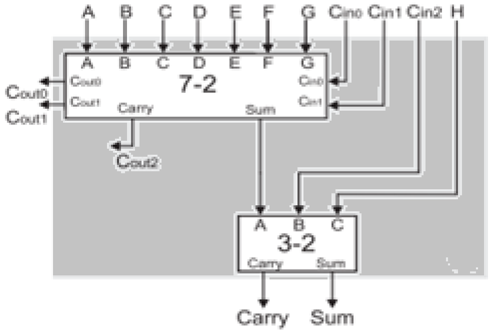


Fig. 6 The structure of 8-2 adder compressor with Combination of 7-2 and 3-2 adder compressors [8]

5 HIGH SPEED ADDERS

For any multiplication algorithm contains three steps but in this summation of partial products is an important step to generate the final result. The performance of the multiplier depends on how fast partial products get added to obtain the final result. Many researchers can work in this area to achieve fast adders. The fundamental adder architecture is a Ripple Carry Adder and further develops number of adders such as Carry look ahead adder, Carry select adder, Carry save adder and Carry skip adder etc. In this ripple carry adder is well known for its regular structure and maximum delay because each step waits for the carry from the previous step. Carry look ahead adder have a minimum delay but area associated with these adders are maximum. Carry skip adder gives the more performance than ripple carry adder but it's consisting of extra hardware circuitry to skip the carry generated. Carry save adder gives the further addition by reducing addition there are number of three into two. The major drawback carry save adder consumes larger area. Further carry select adder uses the two ripple carry adders and it doesn't wait for previous stage to execute. The carry select adder with higher bits exhibits excellent area and speed trade off compare with other adder architectures. Many modifications can be dined in carry save adder for sacrificing its speed for area. To implement a *W*-bit 3-2, 4-2, 5-2 and 7-2 adder compressors it is needed a recombination of partial *Carry* and *Sum* terms. To make the recombination of *Carry* and *Sum* it is used a cascade of half-adder and full-adders circuits in a Ripple Carry form, as presented in the example of the figure 7, for an 8-bit 4-2 compressor. In this work we use a more efficient Carry Look Ahead adder (CLA) to perform the recombination of the partial results. We have used a pipeline stage between the line of the compressors and the adder line used to recombine the partial results (highlighted in figure 7).The design of 16-Bit high adder using different 8-2 adder compressors is shown in Fig. 8. The main objective in this section we can design high speed adders using four 8-2 adder compressors and already explained in section 4

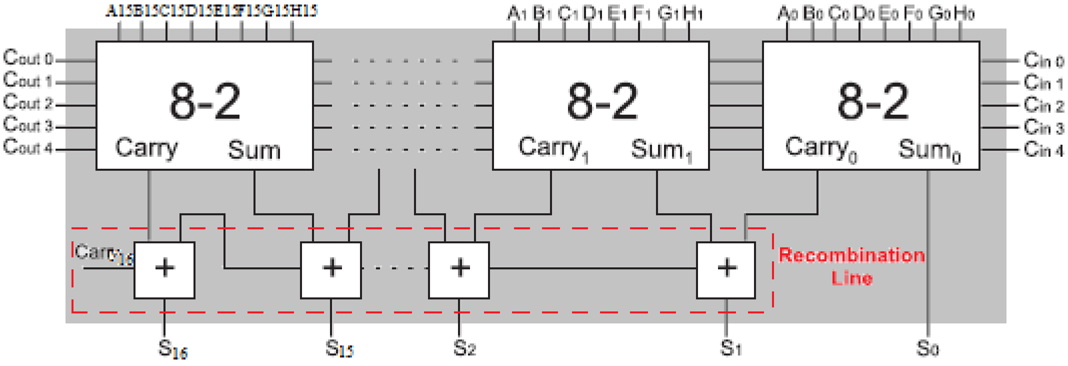


Fig. 8: 16- bit adder using 8-2 ACs[9]

6 PROPOSED APPROXIMATE MULTIPLIER

The Proposed Multiplier is shown in Fig. 9. In this architecture, we can use above section 16-bit adder using 8-2 ACs at partial production and final sum stage after complete the process of OR gate operation. Implementation above multiplier develops the Verilog code and then simulation we can use Xilinx 14.7.

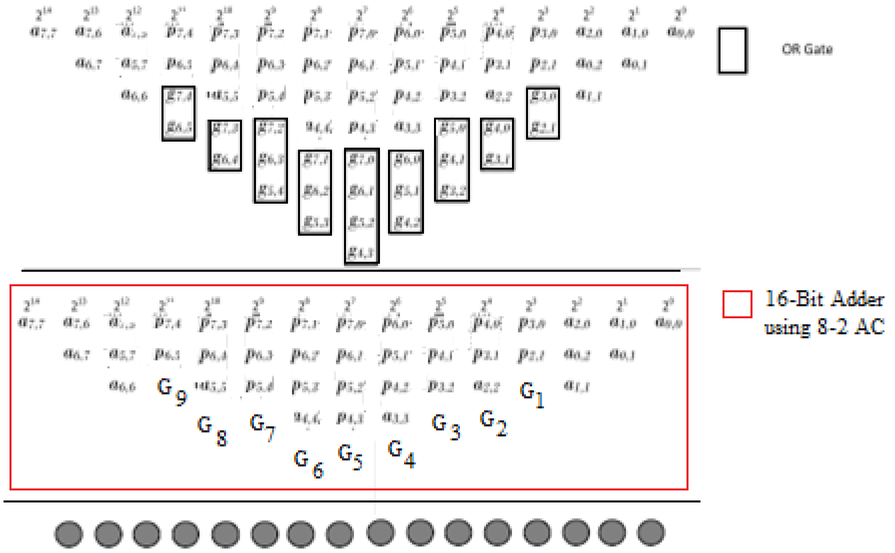
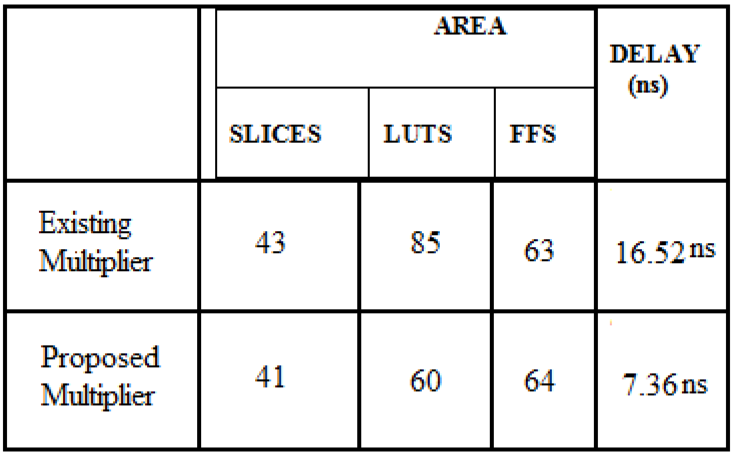


Fig. 9 The Proposed approximate Multiplier with 16-Bit adder

7 RESULTS AND DISCUSSION

The design was synthesized on Xilinx ISE and the functional verification of approximate existing and proposed multiplier was done on Xilinx ISIM. The targeted device is of Spartan-3e of Spartan family. The grade speed of the design is set to -5. The following section contains the results obtained by synthesizing the design in Xilinx ISE 14.7. The comparison result is shown in Table 1 and resultant comparison plot is shown in Fig. 10.



	AREA			DELAY (ns)
	SLICES	LUTS	FFS	
Existing Multiplier	43	85	63	16.52 ns
Proposed Multiplier	41	60	64	7.36 ns

Table 1 Comparison of Existing and Proposed Approximate Multiplier in terms of Area and delay

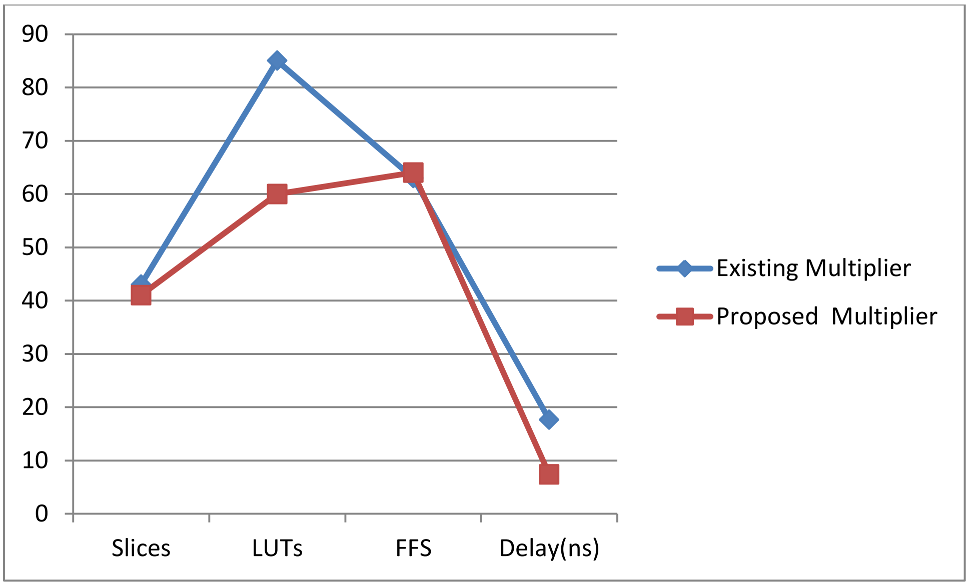


Fig. 10 Performance analysis of Existing and Proposed Multiplier

8. CONCLUSION

Proposed multiplier increases the Speed so that it can multiply and used in image processing applications. It is highly efficient, 55.44 % delay reduced compared to that of previous works. Adaptive voltage level at ground reduces the power consumption by lifting-up the ground potential whenever required and decreasing the voltage through transistors whenever needed. So this multiplier architecture can be used for high speed applications like data mining, and also the cases where the error occurrence is not a major concern.

REFERENCES

[1] V. Gupta, D. Mohapatra, A. Raghunathan, and Roy, “Low-power digital signal processing using approximate adders,” *IEEE Trans. Comput.Aided Design Integr. Circuits Syst.*, vol. 32, no.1,pp.124 137, Jan.2013.

[2] A.Momeni,J.Han, P.Montuschi,and F.Lombardi, “Design and analysis of approximate compressors for multiplication,”*IEEETrans.Comput.*, vol. 64, no. 4, pp. 984–994, Apr. 2015.

[3] S.Narayanamoorthy, H.A.Moghaddam, Z.Liu, T.Park, andN.S.Kim,“Energy-efficientapproximatemultiplication for digital signal processing and classification applications,” *IEEETrans. Very Large Scale Integr. (VLSI) Syst.*, vol.23, no. 6, pp. 1180–1184, Jun. 2015.

[4]G.Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, “Design-efficient approximate multiplication circuits through partial product perforation,” *IEEE Trans. Very Large Scale Integr.(VLSI) Syst.*, vol. 24, no. 10, pp. 3105–3117, Oct. 2016.

[5]C. Liu, J. Han, and F. Lombardi, “A low-power, high-performance approximate multiplier with configurable partial error recovery,” in *Proc.Conf.Exhibit. (DATE)*, 2014, pp. 1–4.

[6]C.H.Lin and C.Lin, “High accuracy approximate multiplier with error correction,” in *Proc. IEEE31st Int. Conf. Comput.Design*, Sep. 2013, pp. 33– 38.

[7]Suganthivenkatachalam and seek-bum KO, senior member IEEE. “Design of power and area efficient approximate multipliers” IEEE transactions on VLSI systems.2017.

[8] J. S. Altermann, E. A. C. da Costa, and S. Bampi, “Fast Forward and Inverse Transforms for the H.264/AVC Standard using Hierarchical Adder Compressors,” in Proc. IEEE/IFIP Int. Conf. VLSI Syst. Chip (VLSI-SoC), Madrid, Spain, pp. 310–315, Sep. 2010.

[9]Bianca Silveira, GuilhermePaim, Cláudio Machado Dinizand and Sergio Bampi, "Power-Efficient Sum of Absolute Differences Hardware Architecture Using Adder Compressors for Integer Motion Estimation Design", IEEE Trans. on Circuits and Systems–I, pp. 1-12, 2017.