International Journal of Scientific Research and Engineering Development-– Volume 6 Issue 4, July- Aug 2023
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

IMPROVEMENT OF ACCURACY OF FIXED-WIDTH BOOTH MULTIPLIERS USING DATA SCALING TECHNOLOGY

K. Saiman*1, D. Syambabu*2,
*1Associate Professor,Dept. of Electronics and Communication Engineering.
*2Assistant Professor, Dept. of Electronics and Communication Engineering.
RISE Krishna Sai Prakasam Group of Institutions, Ongole, Andhra Pradesh, India.

ABSTRACT—To lessen truncation errors, we suggest a data scaling technique (DST) in the fixed width booth multiplier (FWBM). The proposed DST improves operations in FWBMs and decreases theamountof superfluous bits in the multiplicand. By incorporating a modified radix-4 encoder into the suggested DST, the truncation errors in FWBMs are decreased. We discovered that the suggested DST-signal-to-noise FWBM's ratio was significantly better than that of the existing DST-FWBM, and that combined with a modified booth algorithm, the DST-FWBMs were able to produce data with high output accuracy and low error rates.

The accuracy of DST-FWBMs was very near to the optimal value of a post-truncated multiplier. To evaluate its effectiveness The accuracy of FWBMs was found to be significantly improved by the proposed DST approach, making this technology appropriate.

KEY WORDS— DST,FWBM, Truncation, Radix-4.

1 INTRODUCTION

Finite impulse response filters and discrete cosine transform (DCT) are two examples of digital signal processing (DSP) algorithms that frequently employ multipliers. It has been debated for a long time how to create a fixed width multiplier that is straightforward but precise for use in DSP systems. The Baugh-Wooley (BW) array multiplier and the Booth multiplier are two of the most well-liked varieties of fixed-width multipliers. Because there are less truncated partial products because to the Booth encoder, Booth multipliers are more accurate than BW multipliers. The statistical likelihood of a low-error fixed-width Booth multiplier and a simulation are used to determine the compensating bias of truncated partial products Although simulation-based error correction is an accurate technique for FWBMs, designing its circuit takes time. To accomplish continuous compensation in FWBMs, the statistical properties from a simulation were extracted and linear regression was applied. employed a curve-fitting method to get adaptive compensation and minimize truncation error. Similar to this, an exhaustive simulation was used to create the circuit for the comparison method, and Wang used more product data to increase accuracy. However, creating an exhaustive simulation takes time, especially for long-width multiplication, even though it produces precise compensation. The calculation of the

likelihood that an element will be present in the truncated partial products of multipliers is done using probability methods, which can shorten simulation times.

2 LITERATURE SURVEY

HighThroughputDA-basedDCTwithHighAccuracyError-compensatedAdderTreeby Y.H.Chen,T.Y.Chang,andC.Y. Li [1]

An error-compensated adder-tree (ECAT) is developed to cope with the truncation errors and to achieve low-error and high-throughput discrete cosine transform (DCT) design by operating the shifting and addition in parallel. In order to achieve peak-signal-to-noise-ratio (PSNR) standards, 9-bit distributed arithmetic-precision is used in this study rather than the 12 bits used in earlier efforts. In order to meet the PSNR requirements specified in the earlier studies, a 1Gpels throughput rate with gate counts of 22.2 K is using an area-efficient DCT core.

Design and analysis of approximate redundant binary Multipliersby W. Liu, T. Cao, P.Yin,Y.Zhu,C.Wang,E.E.Swartzlander,andF.Lombardi [2].New strategies for computational efficiency have been developed as technological scaling reaches its limits. When utilized in error-tolerant applications, approximate computing is a promising technique for high performance and low power devices. A substantial amount of study has been done on approximate arithmetic designs for approximate circuits. The design of approximate redundant binary (RB) multipliers is examined in this study. For the RB multipliers, two approximative Booth encoders and two RB (full and half) adder-based RB 4:2 compressors are proposed. By taking into account the error properties of both the approximate Booth encoders and the approximate RB compressors, the approximate design of the RB-Normal Binary (NB) converter in the RB multiplier is also investigated.

Low-error and hardware-efficient fixed width multiplier by using the dual-group minorinputcorrectionvectortolowerinputcorrectionvectorcompensationbyI.C.WeyandC.C.Wang [3]

In order to reduce input correction vector compensation error, we suggest a new error-compensating circuit using the dual group minor input correction vector. The hardware complexity of the error compensation circuit can be reduced by making use of the minor input correction vector's symmetric nature. The hardware complexity only marginally rises as the multiplier input bits increase since the error correction circuit is primarily built from the "outer" partial products. The truncation error can be reduced in the suggested 1616 bits fixed-width multiplier by 87% when compared to the direct-truncated multiplier, and the transistor count can be decreased by 47% when compared to the full-length multiplier.

Low-errorfixed-widthBoothmultipliers by M.A.Song,L.D.Van,andS.Y.Kuo [4].

We suggest two fixed-width 2's complement Booth multipliers that can combine an n-bit multiplicand with an n-bit multiplier to produce an n-bit product. Our multipliers are smaller in size, take up less space, and have shorter critical path delays than earlier designs. In order to build a multiplier suitable for VLSI implementation, a four-step process is used to look for the best bias for error-compensation. Last but not least, we implement our solutions in a voice signal processor to demonstrate their improved performance. According to simulation studies, the precision of the new design is superior to that of the fixed-width Booth multiplier that came before it. Only a few logic gates are needed to produce an average error reduction of 65-84% when compared to a direct-truncation fixed-width multiplier.

3 EXISTING SYSTEM

3.1BOOTH MULTIPLIER

The Booth multiplier, in contrast to normal multipliers, is a capable multiplier that dependably handles both positive and negative numbers. The majority of the time, duplication is accomplished by add-and-shift operations, where each multiplier bit creates a unique piece of the multiplicand that must be added to the fractional item. When the multiplier is larger in size, the multiplicand must be multiplied by a greater amount, which causes a high deferral. since the amount of expanded activity determines the deferral. Limiting expansion will help us achieve better execution by reducing the number of incomplete items. Booth calculation is the productive calculation that will restrict the amount of multiplicand.

We partition "y" into three bit segments that overlap by one bit in order to compute the product x*y using the radix-4 Booth multiplier. For y-1 = 0, the final zero on the right has been added. Each group provides a yk value, which specifies an operation. A dual-word size register that stores the sum after each operation is finished is used to compute the product. The encoded values' meaning is condensed in Table I.

Y_{2k+1}	Y_{2k}	Y_{2k-1}	E_k	*Effect on sum*
0	0	0	0	Add '0'
0	0	1	+1	Add 'x'
0	1	0	+1	Add 'x'
0	1	1	+2	Shift 'x' left ,add
1	0	0	-2	Take2's(x),Shiftleft and add
1	0	1	-1	Add 2's(x)
1	1	0	-1	Add 2's(x)
1	1	1	0	Add '0'

Table3.1.1 Boothrecordingtable

3.2 BOOTH ALGORITHM

Following are the steps to implement the Booth Algorithm:

Forming the Booth recoding table is the first step.

Think about the multiplier in three-blocks, where each block overlaps the one before it, and create the table using Table3.1.

Booth Algorithm in Step 2

According to the Booth rule, the multiplicand can be shifted after being added to, subtracted from, or left unaltered from the partial product.

When compared to the standard multiplication method, the Radix-4 booth encoding reduces the amount of multiplier bits, which also minimizes the number of partial products. The length of the multiplier is determined by this encoding procedure.

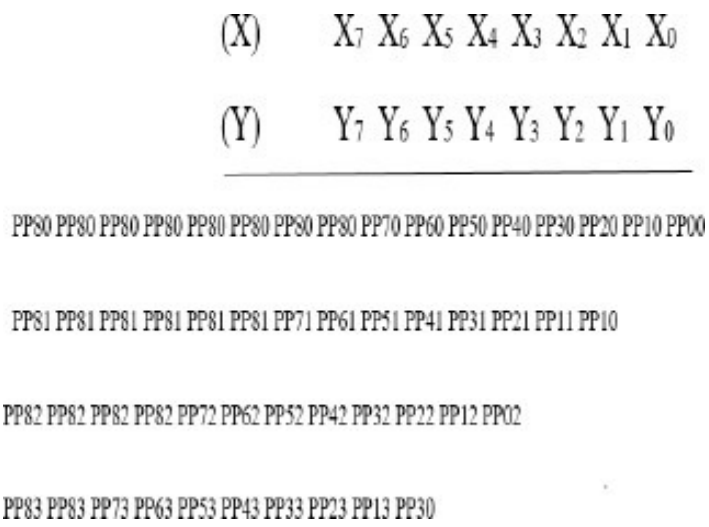


Fig.3.2.1 Generation ofpartialproductsforRadix-4 multiplier

These registers, pp, and goods start off empty. The first partial product is created and then saved in pp (register). The partial product in this case has a length of m, which must be increased by sign extinction and transformed to (m + n) bitsThese registers, pp, and goods start off empty. The first partial product is created and then saved in pp (register). The partial product in this case has a length of m,

which must be increased by sign extinction and transformed to (m + n) bits.

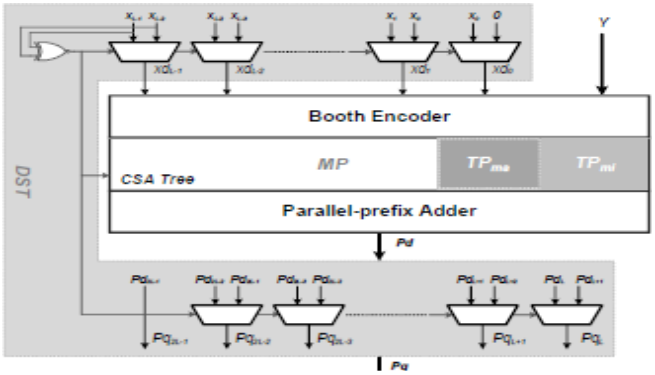


Fig.4.1.1 Architectureofproposed DST-FWBMwith DSb

4 PROPOSED SYSEM:

Booth encoding maps three concatenated inputs, y2i+1, y2i, and y2i1, to yi in order to decrease the number of partial products. The value of the 1-bit nonzero code nzi relies on whether yi is equal to zero. As a result, the partial product array's rows can be expressed as Q = L/2, where L is an even number. The partial results of a L L FWBM can be depicted as shown in Fig. 1 based on the encoding of yi. The main part (MP), which contains the most significant L columns, and the truncation part (TP), which contains the least significant L columns, can be separated to create the partial product array.

4.1 PROPOSED DST-FWBM

TheproductPcanbe expressedasfollows:

P= MP+TP.(2)

Forthefixed-widthmultiplication,theproductP istruncatedtoPq:

$$P \approx P_q = MP + TP$$
$$= MP + \sigma \cdot 2^L$$

Ingeneral,the2L-bitproductPcanbeexpressedusingatwo'scomplementrepresentation.

$$X = -x_{L-1}2^{L-1} + \sum_{i=0}^{L-2} x_i \cdot 2^i$$

$$Y = -y_{L-1}2^{L-1} + \sum_{i=0}^{L-2} y_i \cdot 2^i$$

$$P = X \times Y$$

4.2 KOGGE STONE ADDER

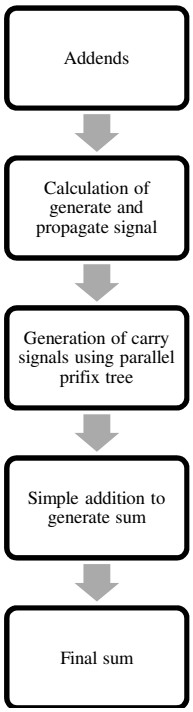
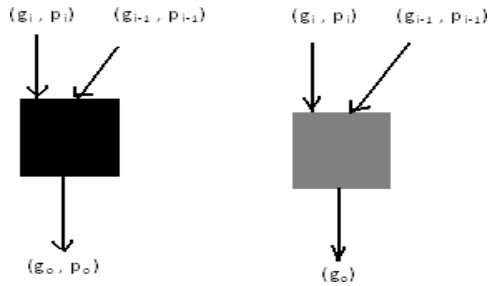


Fig.4.2.1 Kogge Stone Adder

MUXs, whereas the DS Module can be produced using L D-to-1 multiplexers (MUXs) in which D = (DSb + 1). As a result, the (2L 1) D-to-1 MUXs are used in the proposed DST circuit, and the area of the total DST-FWBM circuit is equal to the sum of the areas of the low-error FWBM and the (2L 1) D-to-1 MUXs. The area of a 3-to-1 MUX is supposed to be twice that of a 2-to-1 MUX, the area of a 4-to-1 MUX is assumed to be three

times that of a 2-to-1 MUX, and so on in order to evaluate the area overhead and accuracy. Thus, a D-to-1 MUX over a 2-to-1 MUX requires a (D 1)-fold increase in area. Area of the projected DST

However, the SNR does not significantly improve when multiplied by the DSb value, and an improvement gap is visible between DSb = 0 and DSb = 1, per the study of the rise in the SNR shown in Table II. Because this value offers the proposed DST the highest level of cost efficiency, we picked DSb= 1 for this investigation. Fig. 5.1 depicts the proposed DST-FWBM circuit's architecture. However, when the proposed DST is employed, a high SNR value is obtained, and the DSb increases.



These final two rows are added using a fast adder known as the Kogge stone adder (KSA) in the suggested multiplier. As seen in Fig. 4.2, the Kogge Stone Adder (KSA) adds the bits in parallel prefix form.

The three stages of the Parallel Prefix expansion are shown in fig1. The convey contribution for each adder is created using the fundamental produce and engender sign.

The sum output is created at each bit I of the two operand block by adding the two input signals (ai and bi) to the corresponding carry-in signal

Sumi = ai^ bi^carryi (1)

is the equation to produce the output's total (1) The most important and time-consuming procedure is the computation of the carry-in signals at every bit. The carry-in signals for individual bit additions are designed with the carry-look-ahead scheme of adders (CLA) in mind. This is accomplished by applying the equations to generate the two signals generate (gi) and propagate (pi).

Gi=ai^bi(2)

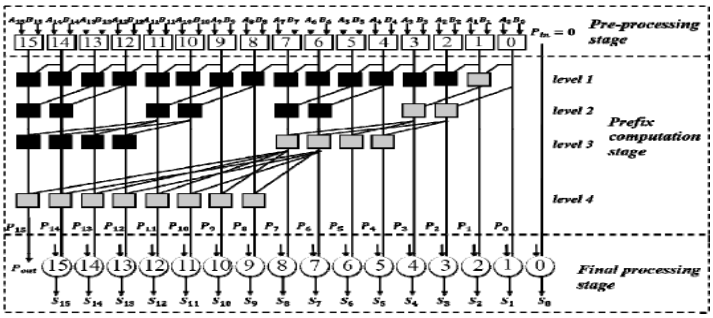
Pi=ai^ bi(3)

Thecarryinsignal foranyadderblockiscalculated byusing theformulaCi+1=giV(pi) (4)

Wherecimustbeexpandedto calculateci+1 atanylevelofaddition

4.3 OPERATORS

Parallel Prefix adders compute carry-in at each level of addition by combining generate andpropagatesignalsinadifferentmanner.Twooperatorsnamel y*black*and*gray*areusedinparallelprefixtreesareshown infig(a),fig(b)respectively.



(a) blackoperator (b)grayoperator

Fig4.3.1 OperatorsusedinParallelPrefixtrees

The black operator computes one set of generate and propagate signals (go, po) using the equations below after receiving two sets of generate and propagate signals (gi, pi), (gi-1, pi-1).

Go=giV(pi^gi-1) (5)

Po =pi^pi-1 (6)

Thegrayoperatorreceivestwosetsofgenerateandpropagatesignals(gi,pi),(gi-1,pi-1),computesonlyonegeneratesignalwiththesameequationasinequation(5).

It is clear that a major benefit of the tree-structured adder is that, for an N-bit wide adder, the critical path caused by the carry delay is on the order of log 2N. The prefix network's configuration results in a number of adder families. to discuss the different carry-tree structures. The modified Kogge-Stone adder, which is renowned for having little logic fanout and depth, is the subject of this study. Here, the grey cell (GC) solely provides the left signal whereas the black cell (BC) generates the ordered pair in equation (1).

It is well known that the interconnect area is considerable, but for an FPGA with significant routing

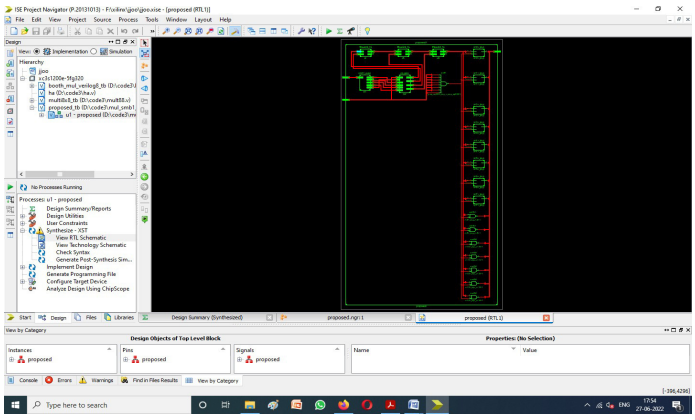


Fig4.3.216-bitKogge-Stone adder

overhead from the start, this is not as significant as in a VLSI implementation. The Kogge-Stone prefix network features built-in redundancy because of its regularity, which has implications for fault-tolerant designs. Study is also conducted on the scarce Kogge-Stone adder. This hybrid architecture allows the carry prefix network to be reduced by completing the summing process.

.

5RESULTS:

5.1SIMULATIONRESULTOFMULTIPLIER

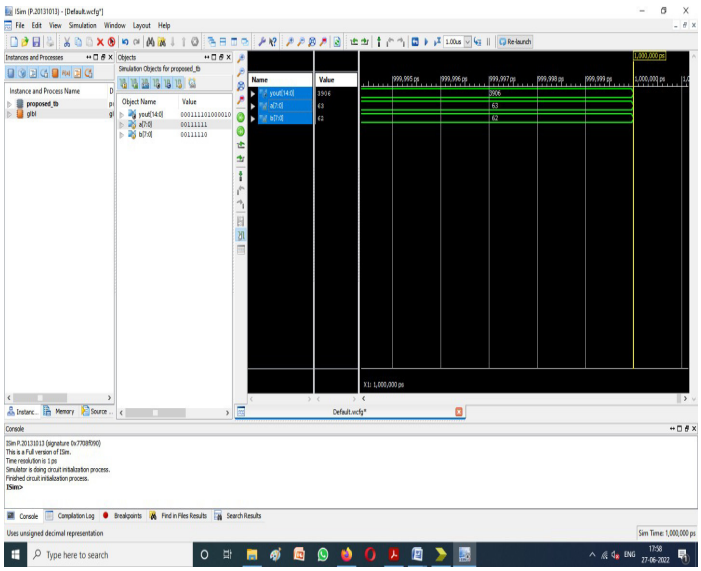


Fig5.1 SimulationResult ofMultiplier

5.2RTLSCHEMATICOFMULTIPIER

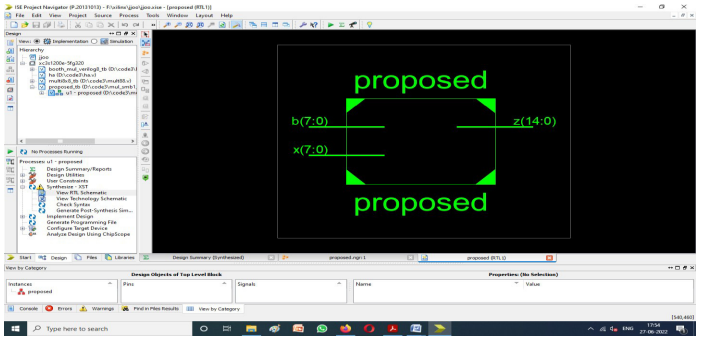


Fig 5.2 RTLSchematicofMultiplier

5.3 ESTIMATIONOFPOWER

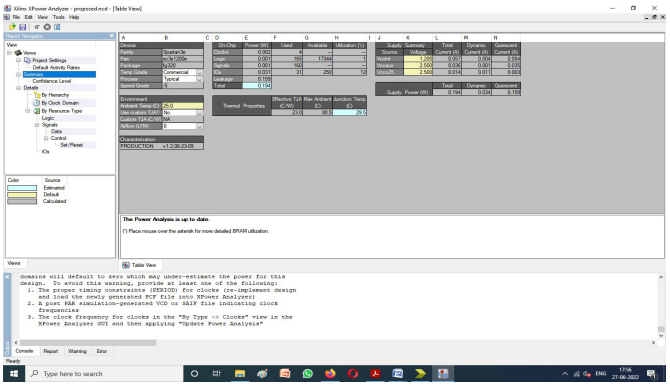
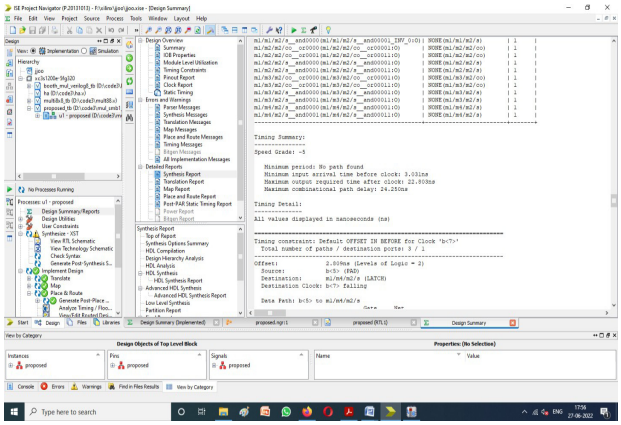


Fig5.3 Estimatespowerusingmultiplier.



5.4 ESTIMATIONOFDELAY

Fig5.4 Estimation ofDelay

6 COMPARISOIN TABLE

	Existingsystem	Proposedsystem
Delay(ns)	38.592	24.250
Speed(MHz)	25.9	41.2
Area	206	236

6.1.1Comparisonbetweenpoweranddelay

7 CONCLUSIONS:

We demonstrated how to apply a DST circuit to low-error FWBMs; the proposed DST circuit significantly increased the FWBMs' accuracy. The accuracy of the suggested DST-FWBM demonstrated a sufficiently low area cost and was rather near to the desired accuracy value of P-T multipliers. The proposed DSTFWBM acquired good accuracy during system application evaluation. The DST circuit further enhanced the long-width FWBMs' accuracy. In conclusion, DST is applicable to DSP applications, especially those that demand great accuracy.

8 REFERENCES

[1] HighThroughputDA-basedDCTwithHighAccuracyError-compensatedAdderTreeby Y.H.Chen,T.Y.Chang,andC.Y. Li.

[2] Design and analysis of approximate redundant binary Multipliers by W. Liu, T. Cao, P.Yin,Y.Zhu,C.Wang,E.E.Swartzlander,andF.Lombardi.

[3] Low-error and hardware-efficient fixed width multiplier by using the dual-group minorinputcorrectionvectortolowerinputcorrectionvectorcompensationby I.C.WeyandC.C.Wang.

[4] Low-errorfixed-widthBoothmultipliers by M.A.Song,L.D.Van,andS.Y.Kuo.

[5] Y.H.ChenandT.Y.Chang,"AHigh-AccuracyAdaptiveConditional-ProbabilityEstimatorforFixed-widthBoothMultipliers,"IEEETrans.CircuitsSyst.I,vol.59,no.3,pp.594–603,Mar.2012.

[6] W. Q. He, Y. H. Chen, and S. J. Jou, "High-Accuracy Fixed-Width Booth Multipliers Basedon Probability and Simulation," IEEE Trans. Circuits Syst. I, vol. 62, no. 8, pp. 2052–2061, Aug.2015.

[7] Y. H. Chen, "An Accuracy-Adjustment Fixed-Width Booth Multiplier Based on MultilevelConditionalProbability,"IEEETrans.VLSISyst.,vol.23,no.1,pp.203–207,Jan.2015.

[8] Z. Zhang and Y. He, "A Low-Error Energy-Efficient

Fixed-Width Booth Multiplier WithSign-Digit-Based Conditional Probability Estimation," IEEE Trans. Circuits Syst. II, vol. 65, no.2,pp.236–240,Feb.2018.

[9] M.Chakraborty,R.Shaik,andM.H.Lee,"ABlock-Floating-Point-BasedRealizationoftheBlock LMS Algorithm,"IEEETrans.CircuitsSyst.II,vol.53,no.9,pp.812–816,2006.

[10] B. Parhami, Computer arithmetic: algorithms and hardware designs. Oxford, UK: OxfordUniversityPress,2000.

[11] S.C.HsiaandS.H.Wang,"Shift-register-baseddatatranspositionforcost-effectivediscretecosinetransform,"IEEETrans.VLSISyst.,vol.15,no.6,pp.725–728,Jun.2007.

[12] J. P. Wang, S. R. Kuang, and S. C. Liang, "High-Accuracy Fixed-Width Modified BoothMultipliersforLossyApplications,"IEEETrans.VLSISyst.,vol.19,no.1,pp.52–60,Jan.2011.

[13] Y.H.Chen,T.Y.Chang,andR.Y.Jou,"Astatisticalerror-compensatedBoothmultiplieranditsDCTapplications,"in Proc.IEEERegion10Conf.,2010,pp.1146–1149.

