IMPLEMENTATION OF LOW POWER AND HIGHACCURACY 2D FIR FILTER USING APPROXIMATEMULTIPLIER FOR IMAGE PROCESSINGAPPLICATIONS

Merfeena[1],ShanubhaMariJero[2],DyanaChristilda[3]

[1]Student, Department of Electronics and Communication Engineering, Vins Christian College of Engineering, Tamil Nadu,India.
[2]Student, Department of Electronics and Communication Engineering, Vins Christian College of Engineering, Tamil Nadu,India.
[3]Professor, Department of Electronics and Communication Engineering, Vins Christian College of Engineering, Tamil Nadu,India.

ABSTRACT

Themultiplieristhemostimportantarithmeticfunctionalunitinmanyapplications,andsincetheseapplicationsfrequently call for numerous multiplications, they consume a lot of power. The multipliers in these gadgets arethe biggest consumersof electricity. Speed, area, andpower are the three keyfactors that influence themultiplier's performance. If the speed increases, a lot of space is consumed, and the opposite is also true. In thisstudy, we provide a roughly multiplier with excellent accuracy and minimal power. The accuracy of theapproximation multiplier is a design parameter for an error-tolerant multiplier (ETM). Compared to existingmultipliers,itutiliseslessspaceandelectricity.WhencomparedtoVedicmultiplier,Wallacetreemultiplier,andboothmultiplier,experimentalresultsdemonstratethatthesuggestedapproximationmultiplieruseslesselectricityonaverage.Inthispaper,weprovideanaccurate,error-tolerantmultiplier.Thesuggestedapproximate multiplier is simulated, synthesised, and targeted on a virtex 6 FPGA device in order to study itsperformance.
Keywords:Approximatemultiplier,Realtimeapplication,Errortolerant.

INTRODUCTION

One of the most potent DSP tools for executing various tasks like filter construction, convolution, FFT, and circularconvolutionisthemultiplier.FIRfilterdesignisoneofthemostoftenutilisedapplicationsinDSP.Incommunication,signalandimageprocessing,andembeddedASICs,multipliersarecrucialcomponents.Inmanyapplications,itservesas the most crucial and basic arithmetical function unit. The process of developing multipliers involves a number ofprocesses, and as a result, they take up more space in the hardware, require more energy, and have an impact onperformance. Artificial intelligence, image recognition, and digital signal processing are the applications. Theseapplications require a lot of multiplications, which uses a lot of electricity. Implementing this type of application isdifficult due to its high-power consumption, especially on mobile devices. Numerous studies have suggested methodsforloweringamultipliercircuit'spowerusage.Theapproximationmultiplierisonemethodusedtolowerthepower

consumptionofamultiplier.Theapproximationmultipliertradesoffaccuracyinfavourofreducedcellsize,timinglag,and increasedpowerconsumption.
Two categories can be used to describe the approximate multiplier. The first form involves managing the multiplier'stemporal path. By utilising dynamic voltage scaling, it is accomplished. When a low voltage is provided to themultiplier, the critical route will take longer to complete. As a result, when the time path is violated, the error occursand produces approximations of the intended results. The second type is utilised to redesign the precise multipliercircuitsby altering themultipliers'operationalcharacteristics.TheWallacetreemultiplier,Vedicmultiplier,andboothmultiplierareafewexamplesofmultipliers.Themajorityofthepreviouslysuggesteddesignsforrebuildingmultipliersare inaccuratecompressorswithm inputsandnoutputs.

Oneofthedigitalfiltersthatisfrequentlyusedindigitalsignalprocessingapplicationsinnumerousindustries,includingimaging, instrumentation, communications, etc. is the finite impulse response (FIR) filter. The FIR filter may beimplemented using programmable digital signal processors (PDSPs). However, in order to implement a large-orderfilter, numerous intricate calculations are required, which has an impact on the speed, cost, flexibility, etc. of standarddigital signal processors. Any processor's performance will only be influenced by its power and latency. To get a CPUthat works well, the power and delay in any processor should be reduced. The multiplier architecture is the one that isutilised the most in CPUs. Only the efficient processor is produced if the multiplier's power consumption and latencyaredecreased.

LITERATUREREVIEW

Theapproximatecompressor-based multiplierforimagesproposedby SwathiKrishnaTU*etal*.is15-4.Heemployeda 15-4 approximation compressor that was created using the Xilinx ISE design suite. In applications for imageprocessing,multiplyingoperationscanbecarriedoutwithlesslatency,lesspower,andlessspace.Comparedtoaccuratemultipliers,approximatecompressor-basedmultipliersofferbettercircuitperformancewiththedrawbacksoferrorrateand averagenormalised errordistancevalues. Butin applications likeimage processing, itisreasonable.

A brand-new approximate multiplier design for digital signal processing is put out by Yue Zhao1 *et al.* He presented anew OR and AND gate-based approximate 4-to-2 compressor in this manner. The space and power consumption canbe significantly reduced with the proposed approximative multipliers. Further testing of the proposed approximatemultipliersisdoneinedgedetectionandimagesharpeningapplications.ThePSNRandSSIMfindingsdemonstratetheapplicabilityofoursuggestedapproximatemultipliersforimage processing.

In this paper, Ihsen Alouani1 *et al.* offers a new architecture that uses accuracy as a design criterion and implements arough parallel multiplier utilising heterogeneous blocks. While improving performance and power trade-offs, theproposed heterogeneous multiplier outperforms the tested circuits in terms of output precision. To increase powerefficiency and performance, approximate computing relies on the range of accepted inaccuracy in the calculationprocess.

RoBA multiplier, an approximation-based multiplier proposed by Reza Zendegani *et al.* is intended for high-speed yetenergy-efficient digital signal processing. Three hardware implementations of the approximation multiplier—one forunsignedoperationsandtwoforsignedoperations—arethetechniqueused.Increaseefficiencyandspeedattheexpenseof a slight inaccuracy.

Sana Mazahir *et al.* provides a probabilistic error analysis of a recursive approximate multiplier. The error probabilityanalysis for recursive approximate multipliers with approximate partial products is employed. We obtain precise errorperformanceevaluation.

Kenta Shirane *et al*. recommends designing approximate multipliers with maximal error-awareness. The methodemployed is a methodology for designing a sequence of approximate array multipliers with varying accuracy, area,power, and delay. The results reveal that our developed multipliers outperform existing approximation multipliers intermsofaccuracy-areaefficiency.

Mukesh Kumar Sukla *et al*. proposes a low-power, low-area approximate multiplier with decreased partial products.This work's method proposes a power and area optimised architecture for an error-tolerant multiplier. By applyingapproximationatthelowerlevelofpartialproduct,atrade-offwiththeerrorcharacteristicismade.Theincreaseinareaandpowerisaccomplishedwithaprobabilityerror(PE)of50%.Upto 957.854MHz,thecrucialfrequencyisreached.

Sunghyun Kim *et al.* introduces an approximate multiplier with good performance and low energy consumption forerror-tolerant applications. It is proposed that an approximate binary multiplier be employed for high-performance andenergy-efficient architecture with acceptable error characteristics. The simulation results show that the suggestedapproximate multiplier can produce considerable improvements in all performance measures while maintaining lowerror metrics.

Weiqiang Liu *et al.* suggests designing and testing approximation logarithmic multipliers for low power error-tolerantapplications. This suggested iterative ALMs (IALMs) method employs a setone adder in both mantissa adders duringaniteration,aswellaslower-part-oraddersandapproximatemirroraddersforthefinaladdition.Thenormalisedmeanerror distance of 16-bit approximation LMs is reduced by up to 18% when compared to traditional LMs with exactunits,andthepower-delayproduct is reducedbyupto 37%.

Pabithra.S *et al.* offer an analysis of an approximation multiplier utilising a 15-4 compressor for error tolerantapplications.The15-4compressor wasbuiltusinganapproximate5-3compressorapproach.Becauseitconsumeslesspower, about 5-3 compressors are employed in four distinct ways in 15-4 compressor. When compared to the actualmultiplier with a tolerable error rate, approximate multipliers produced better outcomes in terms of power, area, andspeed.Becausetheprojectedmultiplier'slatency wasinverselytiedto itsspeed,italsoprovided lowerlatency thantheactual one.

EXISTINGMULTIPLIERS

MULTIPLIERS

The execution time of multiplication operations is crucial in arithmetic operations, and it is a major factor in a design'sperformance evaluation. As a result, efficient multipliers for speed, power consumption, and area are required. Thesemultipliers are appropriate for a wide range of high-speed, low-power, and compact VLSI applications. A multiplier isakeybasicbuildingcomponentinthedesignofsystemsthatusedigitalsignalprocessingandotherapplications.Manyacademics are always attempting to build multipliers with low power consumption, high speed, and regular structure,so that they take up less space for compact VLSI implementation. Many algorithms have been presented in the past todo multiplication. Each algorithm has its own set of advantages and disadvantages, as measured by their speed, powerconsumption,andcircuit complexity.

VEDICMULTIPLIER

Multipliers are essential components of digital systems and play a key role in digital design. Vedic mathematics is asystemofmathematicalrulesthatallowsformoreefficientspeedapplication.UrdhvaTiryakbhyamSutrasandNikhilamSutras areutilisedinVedicmultipliers.Inourproject,theUrdhvaTiryagbhyamSutraistakenintoaccountintheVedicmultiplicationmethod.UrdhvaTiryagbhyamservesasthefundamentalsutraforquickandeasymultiplicationprocedures.Theinputdataisdividedinto two equal pieces,followedbythe crossandvertical product.

VEDICSERIESOFMULTIPLIERS

Vedic mathematics is an ancient mathematical system that existed in India. Basic arithmetic methods are powerful,straightforward, andlogical inthis approach.Another advantage is its consistency. These benefits make Vedicmathematics an essential research area. The rules of Vedic mathematics are primarily based on sixteen sutras. UrdhvaTriyakbhyamsutrasand Nikhilamsutrasare employed formultiplication among thesesixteensutras.

When compared to conventional multipliers, Vedic multipliers are considered to be among the best, and UrdhvaTriyakbhyam sutra-based multiplication is more efficient than Nikhilam sutra-based multiplication. Because of itsregularity and simplicity, Vedic mathematics is simple to implement on FPGA. All partial products required formultiplication are calculated far in advance of the actual multiplication. This is a significant advantage of thismultiplication.BasedonaVedicmathematicstechnique,thesepartialproductsarejoinedtogeneratethefinalproduct,resultinginanextremelyfastapproach.MultiplierdesignsbasedonVedicmathematicsarefastandrequirelittlepower.

Multipliers are fundamental and essential components of a digital signal processor. Multiplication is a critical step inincreasingtheprocessingspeedofdigitalsignalprocessors.Multiplierblocksareusedinconvolution,FastFourier transforms, and other transforms. Urdhva Tiryagbhyam is one of the most efficient multiplication procedures in Vedicmathematics.

MULTIPLICATIONUSINGVEDICMULTIPLIER

UrdhvaTiryagbhyamiisauniversalmultiplicationformulathatcanbeappliedtoeverycircumstanceofmultiplication.When the Vedic multiplication method is used, the input data is divided into two equal pieces, followed by the crossand vertical product. The graphic below depicts the 4multiplication process utilising Vedic method. Each square blockrepresentsamultiplier oftwo.Thefirst22multiplier'sinputsareA1A0andB1B0,asindicated.Theleftmostblockisa22multiplierwithA3A2andB3B2 inputs.Themiddletwo22multipliers'inputsareA3A2andB1B0,andA1A0andB3B2.

$$\begin{array}{cc} A_3A_2 & A_1 \\ XB_3B_2 & A_0B_1 \\ & B_0 \\ \hline \end{array}$$

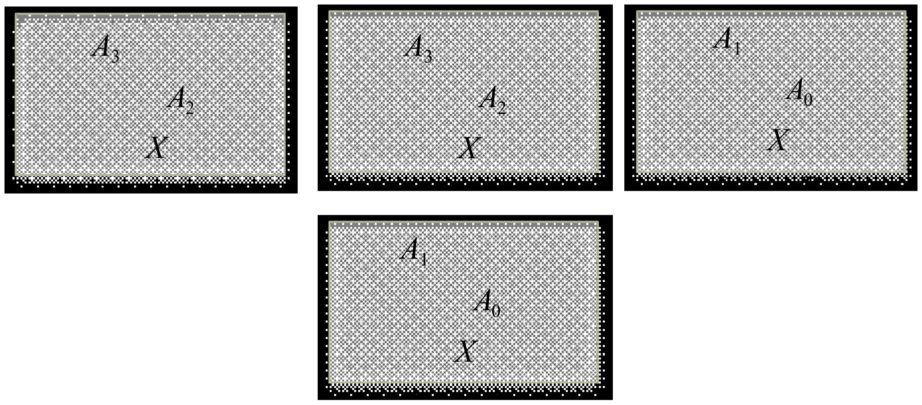


Fig1.4×4multiplicationoperationusingVedictechnique

S33 S32 S31 S30 is produced by multiplying A3A2 and B3B2.S23 S22 S21 S20 is produced by multiplying A3A2 and B1B0.S13 S12 S11 S10 is obtained by multiplying A1A0 and B3B2.S03S02S01S00isobtainedbymultiplying A1A0andB1B0.
Allpartial productscomputedby these four blocksareadded together.

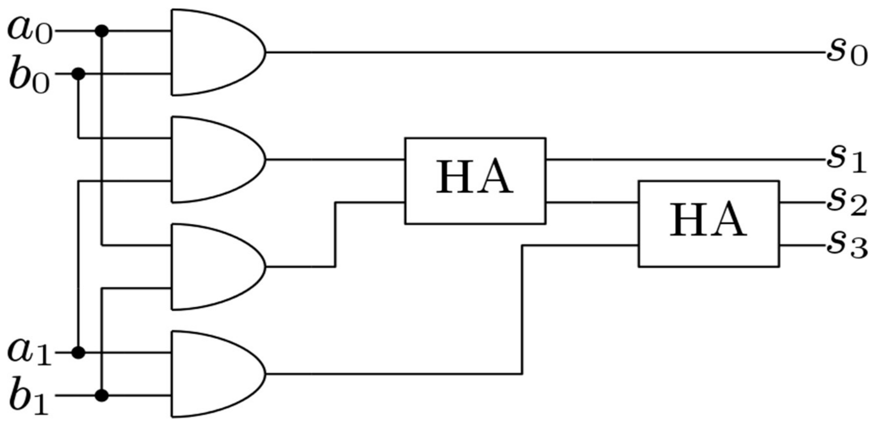


Fig2.2bitVedicmultiplier

BOOTHMULTIPLIER

TheBoothmultiplicationmethod isimportantindevelopingsignedmultipliersthatusemultiplier encodersand reducethe number of intermediate products. Multiplier is the fundamental and essential component of an arithmetic unit inhigh-performance tasks such as digital signal processing (DSP), digital image processing (DIP), and high-performancecentralprocessingunit(CPU).
Using a booth encoder to limit the number of partial products created during the multiplication process is a practicalandefficienttechniqueofenhancingthemultiplier'sspeed.Whenusingtheboothencodingapproach,lessadditionsare requiredthanwhenusingthetraditionalmultiplicationrule.TheBoothEncoder(BE),theBoothSelector(BS),andtheaddertreesummationarethethreeessentialcomponentsofaconventionalBoothMultiplier.Thecarrylookaheadadderisemployedinthiscase.Ithasbeendiscoveredthatthenetarchitecturegeneratedbytheadderandmultiplieroptimisesspeedand area byreducing thenumberofpartialproductsnecessary and loweringtherequired powerconsumption.

MULTIPLICATIONUSINGBOOTHMULTIPLIER

Weconsiderthebitsof'1'inthemultiplier,thenperformthemultiplicationoperationforthosesubsequentmultiplicandbits,andthendisplaythesebits.Ifthebitis'0,'weobtainzeroforthemultiplicandbits,whicharethendisplayedintheiteration steps. As a result, multiplication is a lengthy procedure because each bit is multiplied, and the multiplicationaction takes longer. As a result, regular multiplication requires more time for the multiplication operation where eachand every bit is multiplied, resulting in a greater number of iteration steps. The latency increases as the number ofiteration steps increases. So, we use a booth multiplier to reduce the amount of iteration steps and downsides. Inmultiplication, the basic booth multiplier is used for both signed and unsigned bits. In the instance of this multipliershiftingprocedure,itisdonedirectlyincertaincasesandindirectlyinothers.Soitisasophisticatedprocedureinwhicheach bit isexamined andthenshiftingoccursinmultiplication.

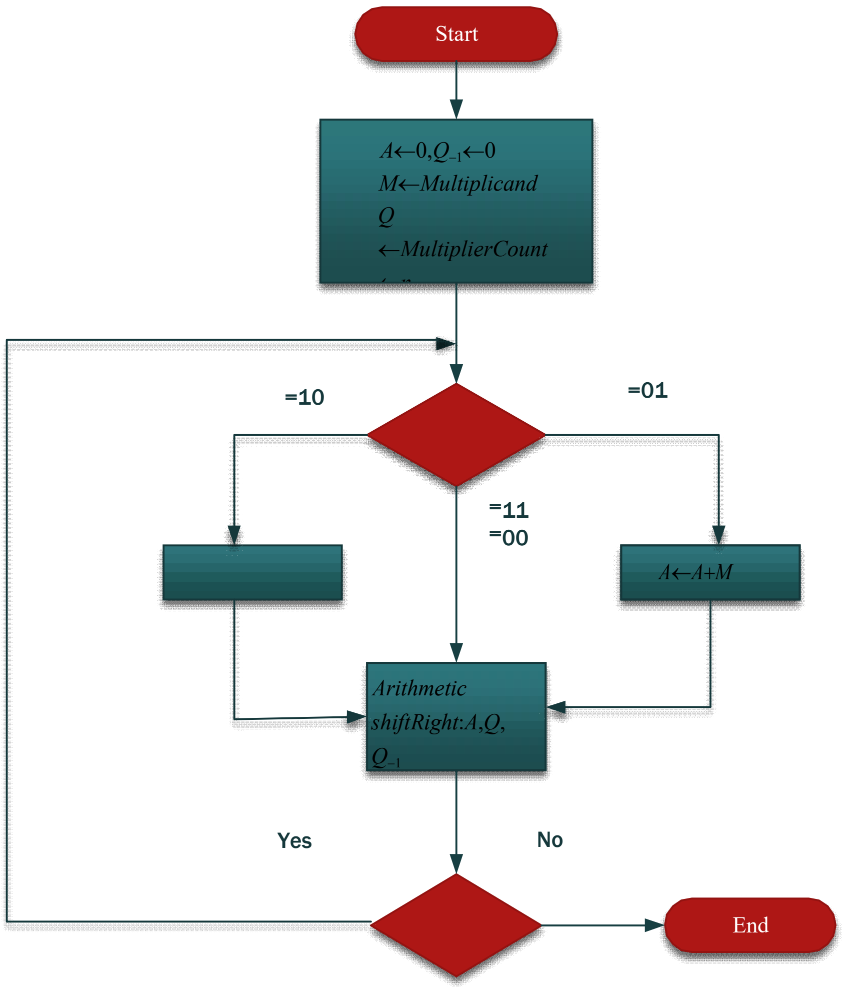

Start
$A \leftarrow 0, Q_{-1} \leftarrow 0$
$M \leftarrow Multiplicand$
Q
$\leftarrow MultiplierCount$
=10
=01
=11
=00
$A \leftarrow A+M$
Arithmetic shiftRight:A, Q, Q_{-1}
Yes
No
End

Fig3.BlockdiagramofBoothmultiplier

The final result comes when the sequence counter approaches zero. As a result, it is considered one of the slowestprocesses to continue and takes a long time to process; as a result, the delay also increases and gradually leads to theincrement of the delay, which then reduces the speed of the multiplier. As a result, electricity consumption rises. Wechoosetheadjustedboothmultiplierafterevaluatingallofthedisadvantages.Whileperformingthemultiplicationwiththeb oothmultiplier, thenumberofiterationsteps willbelowered.

ARCHITECTUREOFBOOTHMULTIPLIER
Complement Generator, Booth Encoder, Partial Product, and Carry Save adder are the four components of thearchitecture.

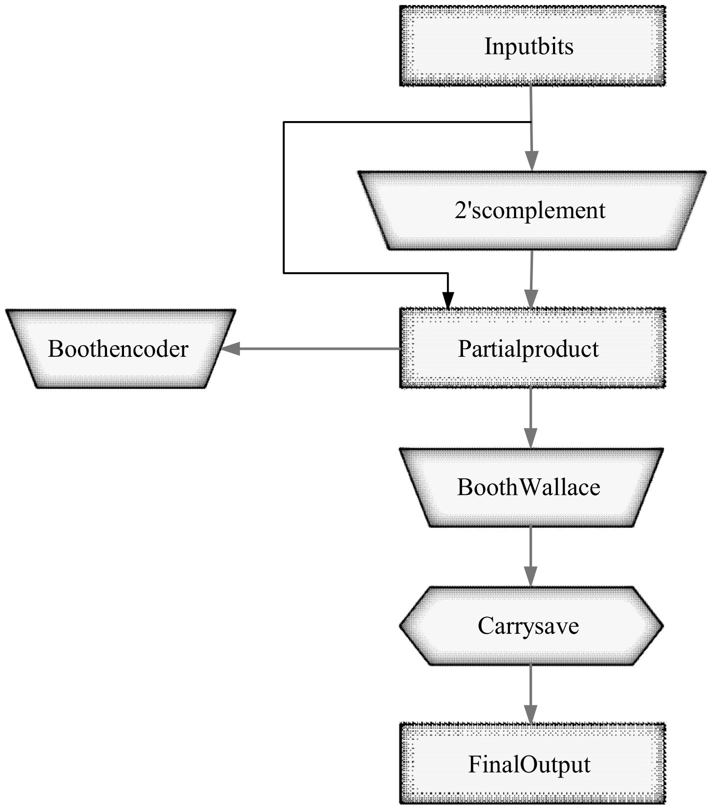


Fig4.Flowchartofboothmultiplier

A) COMPLEMENTCOMPARATOR

In the case of this complement comparator, the multiplicand or the given data is used to generate the 2's complement,andthecomplementedresultsareobtained.Thesesupplementedresultsareusedwhenthereisarequirement;otherwise,thedirect result isusedinsome circumstances accordingtopresetcases.

B) ENCODER

Toperformthemultiplication,thebeginningbitsaregiventotheencoder,andthentheencoder'sappliedbitsaretreatedasone bit forthetwobits,andfinally the multiplicationis performed onthe bits.

C) PARTIALPRODUCTGENERATOR

The decoded bits are obtained at the partial product generator, resulting in the development of fewer partial productsduringnumbermultiplication.Asa result,the quantityofpartial products isreduced.

D) WALLACEMULTIPLIER

The Wallace multiplier also functions as an array multiplier. This multiplier employs half adder and full adder adders.During the multiplicationoperation,everybitis multiplied byevery otherbit.

E) CARRYSAVEADDER

Because the fast addition of partial products is performed and the result is obtained so quickly, this adder is preferredabove other adders. When compared to regular multiplication, the Booth multiplier finds the operand that functions asmultiplier and does multiplication for the algorithm since it reduces the number of steps while completing addition. Inthe case of multiplication, the operation is conducted for each bit of the multiplier with the multiplicand, and then theformationofpartialproductsoccursin theappropriateorder,followedby theadditionofallpartialproductsobtained.

The most intriguing aspect is that the additions made in this multiplication are data dependent, making this a perfectalgorithm. Multiplication of signed numbers is not achievable in the same way that it is for unsigned numbers becausesigned numbers in 2's complement form cannot yield the exact result if the same multiplication technique is used forunsigned numbers. As a result, the booth algorithm is applied, which reduces the final result's sign. Thus, the boothmethod achieveshigh-speedmultiplicationandhasapplicationssuchasdigitalsignalprocessingand radar.

BOOTHALGORITHM

1. Addinga'0'bittotheLSBofthemultiplierandconsideringfromtherightmostofthemultipliertocombinetwobitsfrom therightsideto theleft sideandrespectivemultiplier.
2. 00:11:no operationisperformed.
3. 01:indicatetheendofthestring 1sbefore multiplying the partialproducts.
4. 10:Beginthe stringof1sbyremovingmultiplicandfrom partialproducts.

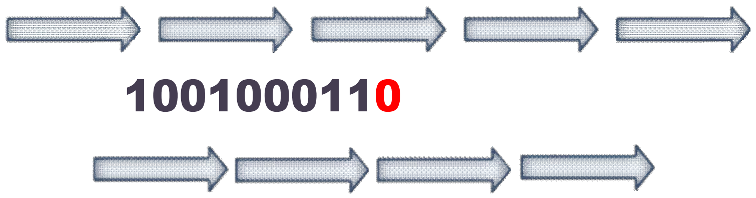


Fig 5.bitcombiningofboothrecorder

The required method of obtaining results for recognising the highest speed multiplier is to enhance parallelism, whichwas effective in obtaining fewer number of subsequent calculation levels. Likewise, the booth algorithm for a radix-4compares three bits using an overlapping technique. Because this multiplication can minimise the number of partialproductsbyhalfwhencomparedto standardmultiplication.

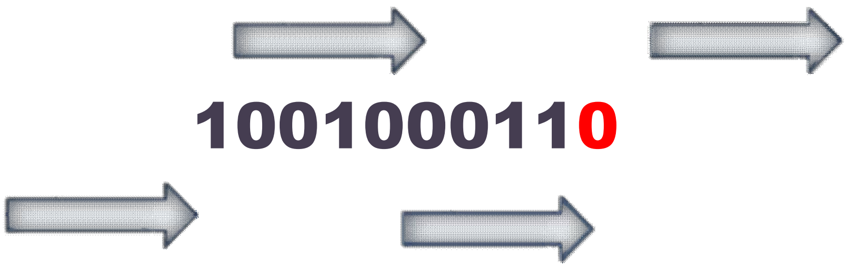


Fig6.Bitpairingasperboothrecorder

Thus,byusingthreebits,thespeedofmultiplicationcanbeincreasedwhilethenumberofmultiplicationstepsiscutinhalf compared to conventional multiplication. Booth multiplication also has several advantages, such as the fact thatwhen three bits are the same, no operation can be performed, reducing the number of adders and the complexity of themultiplier. This multiplier has a distinct operation for successive bit operations and does not require addition andsubtraction foreachstepofmultiplication.

Also,themultiplicationofsignednumbersisnotpossibleassameasunsignednumbersbecausethesignednumbersin2'scomplementformcannotgivetheexactresultifthesameprocessofmultiplicationisappliedforunsignednumbers.That is why booth algorithm is used and deteriorates the sign of the final result. Thus, booth algorithm performs highspeed multiplication and it find itswayin different useslikedigital signalprocessing,radaretc.

WALLACEMULTIPLIER

During1965,computerscientistLuigiWALLACEdesignedtheWALLACEhardwaremultiplier.WALLACEmultiplier is a parallel multiplier that has been extracted. It is slightly faster and necessitates fewer gates. The parallelmultiplier employs a variety of techniques. The WALLACE system is a parallel multiplier scheme that effectivelyminimises the number of adder stages required to complete partial product summing. Wallace multiplier is created byreducing the number of rows in the matrix number of bits at each summation stage using full and half adders. Despitethe fact that WALLACE multiplication has a regular and less complex structure, the operation is slower owing to theserialmultiplicationmethod.Furthermore,theWALLACEmultiplierislessexpensivethantheWallacetreemultiplier.As a result, the WALLACE multiplier is created and analysed by taking into account the many techniques of applyingcomplete adders including distinct logic types (a) Wallace Multiplier Implementation (b) The WALLACE multiplieralgorithm isbased on amatrixstructure. Thematrix'spartialproductisproduced in thefirststepby ANDstages.

WALLACETREEMULTIPLICATION

TheWallacetreeisalengthymultiplicationvariant.Thefirststepistomultiplyonefactor'sdigits(perbit)bytheotherfactor's digits. Each of these partial products has the same weight as the sum of its elements. The weighted sum of alltheseparatial productsyieldsthe final product.

As previously stated, the first step is to multiply each bit of one number by each bit of the other, which is performedusing a simple AND gate, yielding bits; the partial. The partial products of bits am by bn bits have weight $2^{(m+n)}$ in thesecondstep.

The resulting bits are reduced to two numbers in the second stage, which is performed as follows: If there are three ormore wiresofthe sameweight, add thefollowinglayer: -

- Insert anythreewireswiththeidenticalweights intoacompleteadder.As aresult,for eachofthethreeinputwires,an outputwireof thesameweightandanoutputwireofgreaterweightwillbe produced.
- Ifyou stillhavetwowiresofthe sameweight,putthemin ahalf adder.
- Connect thelastwireto thenextlayer ifthere isjustone left.

The final step is to feed the two resulting numbers to an adder, which produces the final

product.Example:

n=4,multiplying$a_3a_2a_1a_0$by $b_3b_2b_1b_0$

1. First,wemultiply everybitbyeverybit:
   - weight1 –a_0b_0
   - weight2 – a_0b_1,a_1b_0
   - weight4 –a_0b_2,a_1b_1,a_2b_0
   - weight8–a_0b_3,a_1b_2,a_2b_1,a_3b_0
   - weight16–a_1b_3,a_2b_2,a_3b_1
   - weight32–a_2b_3,a_3b_2
   - weight64–a_3b_3
2. Reduction layer1:
   - Passtheonlyweight-1wirethrough,output:1weight-1wire
   - Addahalfadderforweight2,outputs:1 weight-2wire,1weight-4wire
   - Addafulladderforweight4,outputs:1weight-4wire,1weight-8wire
   - Addafulladderforweight8,andpasstheremainingwirethrough,outputs:2weight-8wires,1 weight-16wire
   - Addafulladderforweight16,outputs:1weight-16wire,1weight-32wire
   - Addahalfadderforweight32,outputs:1weight-32wire,1weight-64wire
   - Passtheonlyweight-64wirethrough,output:1weight-64wire

3. Wiresat theoutput ofreductionlayer1:
   - weight1 –1
   - weight2 –1
   - weight4 –2
   - weight8 –3
   - weight16–2
   - weight32–2
   - weight64–2
4. Reduction layer2:
   - Addafulladderfor weight8,and halfaddersfor weights4,16,32, 64
5. Outputs:
   - weight1 –1
   - weight2 –1
   - weight4 –1
   - weight8 –2
   - weight16–2
   - weight32–2
   - weight64–2
   - weight128– 1
6. Group thewiresinto apairofintegersand anadderto addthem.

STEPSINVOLVEDINWALLACETREEMULTIPLIERSALGORITHM

• Divideeachbitofoneoftheargumentsbyeachbitoftheother,gettingNoutcomes.Thewirescarryvaryingweightsdepending onthe positionofthe multipliedbits.
• Reduce the amount of partial products to two full adder layers. Divide the wires into two numbers and add them withastandardadder.

WALLACETREEMULTIPLIERUSINGRIPPLECARRYADDER

Ripple Carry Adder is a way for doing a bigger number of additions with the carry ins and carry outs that will belinked. As a result, the ripple carry adder employs many adders. To add multiple-bit numbers, a logical circuit can bebuilt utilising numerous complete adders. Cin, the Cout of the previous adder, is input by each full adder. This type ofadder is known as a ripple carry adder because each carry bit "ripples" to the next complete adder. Take any threevalues with the same weights and feed them into a complete adder. As a result, the output wire will be the sameweight.

• At the first stage, a partial product is formed after multiplication. The data is collected using threewiresand addedusingadders,and thecarryof each stageiscombinedwiththe nexttwodatain thesame stage.
• Usingthe sameprocess,partialproductsarereduced totwo layersoffulladders.
• Atthe finalstage,the sameripplecarryaddermechanismisused, yieldingproducttermsp1 to p8.

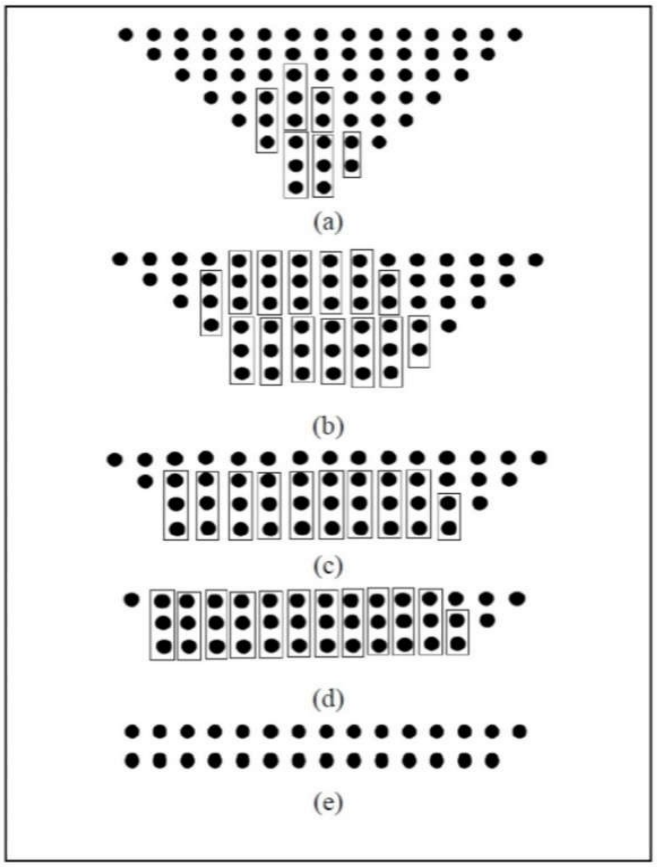


Fig7.Columncompressionschemefor8x8WallacemultiplierAPPRO

XIMATEMULTIPLIER

Approximate circuits are considered as an error-tolerant applications that can tolerate some loss of accuracy withimproved performance and energy efficiency. Multipliers are the key arithmetic circuits in many of such applicationssuch as digitalsignal processing(DSP).

ROUNDINGBASEDAPPROXIMATION

Themainideainthe proposedapproximatemultiplier istomake useoftheease ofoperation whennumbersaretwotothe power n (2n). To elaborate on the operation of the approximate multiplier, first, let us denote that the roundednumbersoftheinput ofAand BbyArandBrrespectively.The multiplicationofAbyBiswrittenas

$$A*B=(A_r-A)*(B_r-B)+A_r*B+B_r*A-A_r*B_r \quad \textbf{(1)}$$

The key observation is that the multiplications of Ar*Br,A*Br and Ar *Br may be implemented just by the shiftoperation.Thehardwareimplementationof(Ar-A)*(Br-B),however,israthercomplex.Theweightofthisterminthefinal result, which depends on the differences of the exact numbers from their rounded ones, is typically small. Hence,we propose to omit this part from equation 1, helping simplify the multiplication operation. Hence, to perform themultiplicationprocess,thefollowingexpressionis beingused

$$A*B=(A_r*B)+B_r*A-A_r*B_r \quad \textbf{(2)}$$

Thus, one can perform the multiplication operation using three shift and two addition/subtraction operations. In thisapproach,thenearestvaluesforAandBintheformof2nshouldbedetermined.WhenthevalueofA(orB)isequaltothe 3* 2p−2 (where p is an arbitrary positive integer larger than one). It has two nearest values in the form of 2n withequal absolute differences that are 2Pand 2p−1. While both values lead to the same effect on the accuracy of theproposed multiplier, selecting the larger one (except for the case of p = 2) leads to a smaller hardware implementationfordeterminingthenearestroundedvalue,andhence,itisconsideredinthispaper.Itoriginatesfromthefactthatt he numbers in the form of 3* 2p−2 are considered as do not care in both rounding up and down simplifying the process,and smallerlogicexpressionsmaybe achievediftheyareusedin the roundingup.zero).

Intheproposedequation,Ar[i]isoneintwocases.Inthefirstcase,A[i]isoneandallthebitsonitsleftsidearezerowhile A[i-1] is zero. In the second case, when A[i] and all its left-side bits are zero, A[i-1] and A[i-2] are both one.Having determined the rounding values, using three-barrel shifter blocks, the products Ar*Br, A*Br and Ar*Br arecalculated. A single 2n-bit Brent-Kung adder is used to calculate the summation of Ar*Br, A*Br. output of this adderand the result of Ar*Br are the inputs of the sub tractor block whose output is the absolute value of the output of theproposedmultiplier.Finally,ifthesignofthefinalmultiplicationresultshouldbenegative,theoutputofthesubtractorwillbenegatedinthesignsetblock.Tonegatevalues,whichhavethetwoscomplementrepresentation,thecorresponding circuit basedonx+1shouldbeused.

Toincreasethespeedofnegationoperation,onemayskiptheincrementationprocessinthenegatingphasebyacceptingits associated error. The significance of the error decreases as the input widths increases. If the negation is performedexactly (approximately), the implementation is called signed MRoBA (SMRoBA) multiplier [approximate SMRoBA(ASMRoBA) multiplier]. In the case where the inputs are always positive, to increase the speed and reduce the powerconsumption,thesigndetectorandsignsetblocksareomittedfromthearchitecture,providinguswiththearchitecturecalled unsigned MRoBA (UMRoBA) multiplier. Shifted to left to generate the final output, an approximate 44 WTMhas been proposed that uses an inaccurate 4:2 counter. In addition, an error correction unit for correcting the rewrittenasoutputshas beensuggested.

To construct larger multipliers, these 44 inaccurate Wallace multipliers can be used in an array structure. Approximatemultipliers are based on either modifying the structure or complexity reduction of a specific accurate multiplier andperformingtheapproximate multiplicationthroughsimplifyingtheoperation.

PROPOSEDMULTIPLIER

PROPOSEDAPPROXIMATEMULTIPLIER

We will discuss the differences between the traditional and proposed multiplication flows in this section. Then wepresent our proposed error correction and high accuracy 4-2 compressor circuit. It then adjusts the multiplier byemployingdynamicinputtruncation.TheoveralldesignofoursuggestedapproximatemultiplierisintroducedinAtlast.

PROPOSEDFLOWOFAPPROXIMATEMULTIPLIER

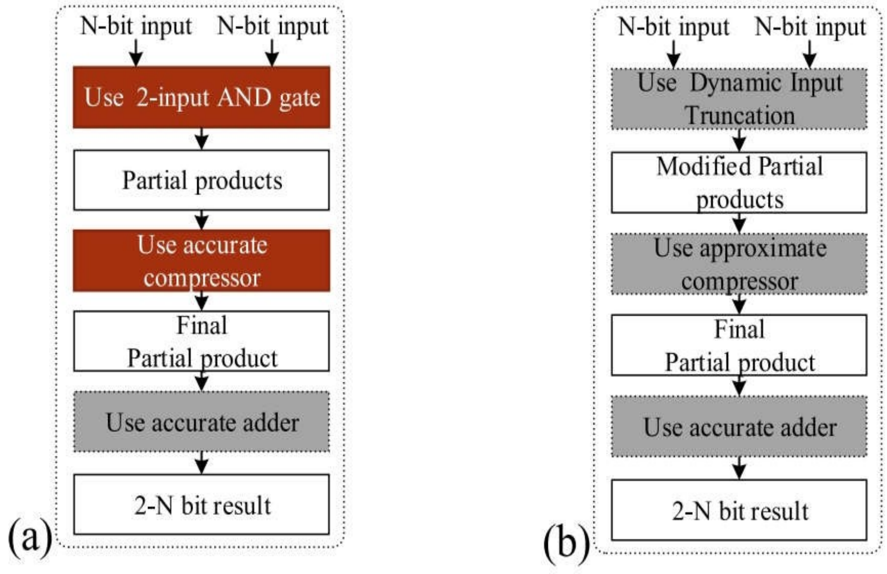


Fig8.(a)traditional(b)proposedmultiplicationflow

This diagram depicts the general customary flow for multiplication that yields an accurate result. The accurate partialproducts are first formed using two input AND gates, and then compressed using precise compressors. Finally, tocompress the results, the accurate adders sum the compressed partial products. The (b) section of the above figuredepicts the projected flow for the proposed approximate multipliers. The steps of creating partial products andcompressing the partialproductsdistinguishtraditionalmultiplicationfromproposedmultiplication.

DYNAMICINPUTTRUNCATION

Dynamic input truncation is a technique for adjusting the accuracy and necessary power. We propose a dynamic inputtruncation strategy that uses the and gate to achieve the adjustable approximate multiplier. The trunc signal conservespower by reducing the PPD in multiplication to zero. Each bit of an 8 8 multiplier corresponds to 8 bits of themultiplicand;thus,weproposeto cut hardware costs bysharing gates withan extra ANDgate.

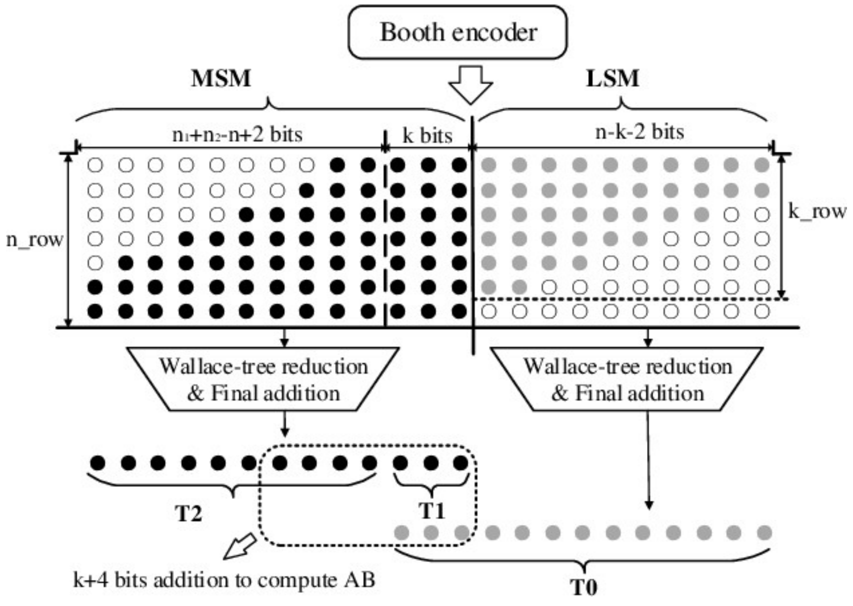

Booth encoder
MSM
LSM
n₁+n₂-n+2 bits
k bits
n-k-2 bits
n_row
k_row
Wallace-tree reduction & Final addition
Wallace-tree reduction & Final addition
T2
T1
T0
k+4 bits addition to compute AB

Fig 9. structure of truncated

multiplierTHEPROPOSEDTECHNIQUEOFAPPROXIMATEMULTIPL IER

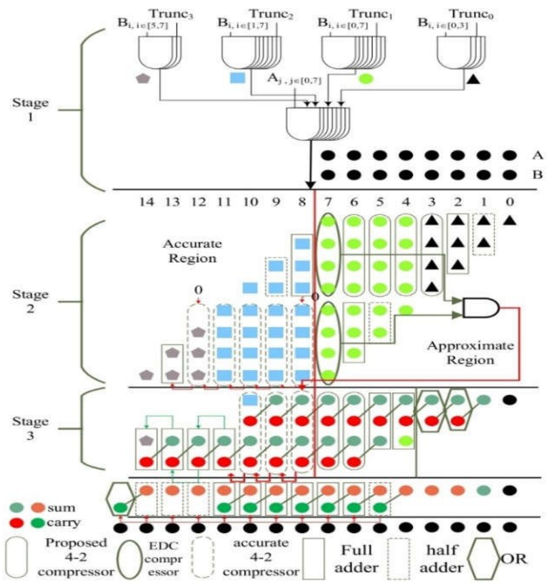


Fig10.Anapproximatemultiplierwiththeproposaltechnique

The figure above depicts an approximation of a multiplier using the proposed technique. Even though the multiplier'sinput width is limited to 8 bits, the proposed technique can be expanded to larger multipliers. There are three steps inthe proposed approximation multiplier. In the first stage, each partial product is generated by two 2-input AND gates,as previously shown, with the gate sharing technique used to cut hardware costs even further. The precision of theresulting partial productcanbedetermined using theshortened signal, depending ontheneeds.

In our proposed approximate multiplier, we design a 4-bit truncated signal with each bit controlling more than onepartial product column, which we call the "3-4-4-4 partition," specifically, each bit from MSB to LSB, correspondingtothe colourskhaki,skyblue,green, andblackin Stage2.

The way the columns are partitioned provides different possibilities for regulating trunc signals, allowing users todynamically adjust the recommended multiplier based on their needs. We ran several experiments to test severalpartitions, and the results reveal that the 3-4-4-4 partition, as well as the 3-3-3-3-3 partition, both maintain a goodcombination of power savings, accuracy, and area overhead. The finer the partition, the more flexible the control overthe amount of power saved and accuracy losses. However, it will suffer from large area overhead in exchange. As aresult,weadoptthe3-4-4-4split throughout all tests.

2DFIRDIGITALFILTER

The2DFIRfilteringisalso calledaslinearspatialfiltering.Itisdesigned using 2Dconvolution.

$$y(m,n)=x(m,n)*h(m,n) \tag{3}$$

$$y(m,n)=\sum_{k1=0}^{k2-1} x(k1,k2)\sum_{k2=0}^{k1-1} h(m-k1,n-k2) \tag{4}$$

wherek1andk2arethepicture'sdimensionsinpixels,h(k1,k2)arethefiltercoefficients,x(m,n)istheoriginalimage,andy(m,n)isthefilteredimage.Figure1depictsthedirectformstructureofa2DFIRfilterwithN=4.Tocreate2DFIRfiltersinhardware,three basicelementsarerequired:anadder,amultiplier,andadelayelement.Themultiplierhasthegreatestinfluenceonspeed,area,andpowerofthesethreefactors.Themultiplier'sspeeddeterminestheexecutiontimeand speedoftheFIRfilterprocessor.

However, the majority of existing architectures are fundamentally based on the traditional multiplier-based design,resultinginhardware-expensivemultipliersaswellasexcessiveenergyusage.Becauseoftheseperformancelimitations,they aretypicallyinappropriatefor embedded systemswithsevere energy efficiency requirements.

The number of multiplications completed in a unit of time is used to determine the performance of the microcontrolleranddigitalsignalprocessor.Asaresult,selectingahigh-speedmultiplierincreasestheperformanceofthe2DFIRfilter.The goal of this project is to create a 2D FIR filter using Vedic Mathematics. Many prior studies concentrated on theimplementation ofFIRfiltersusingvariousmultipliers.

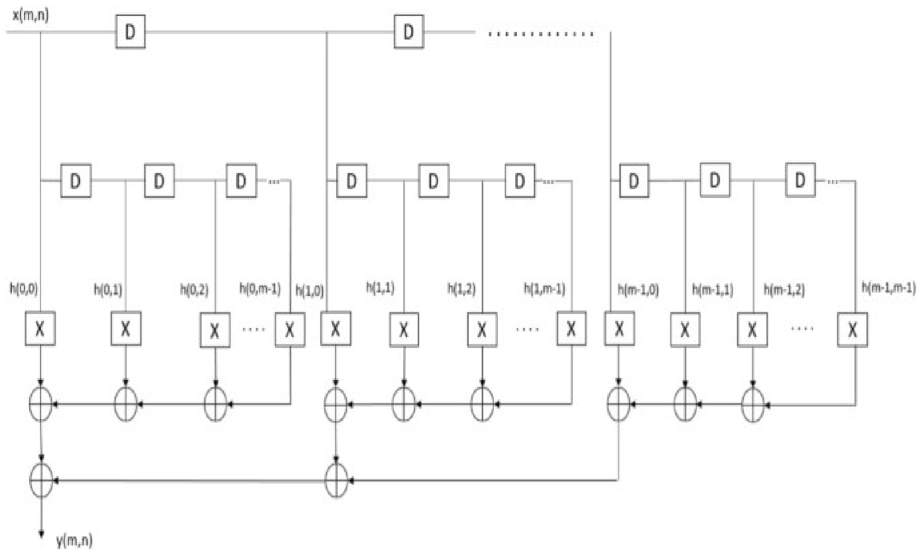


Fig11.2DFIRfilterforimageprocessingapplications

RESULTSANDDISCUSSION

Low power, high speed, and space efficient circuits are preferred for multipliers in the VLSI era. Arithmetic circuits,adders, and multipliers are essential in the design of any signal processing of medical imaging applications. Theperformanceof theaddersandmultipliershasasignificantimpactontheoverallperformance ofthecircuits.For high-performanceapplications,efficientadderandmultipliercircuitsarerequired.Weofferaspeed,area,andpowerefficientapproximate multiplier architecture that is appropriate for signal processing, multimedia processing, machine learning,scientificcomputing,andotherapplications inthis work.

This paper proposes an approximate multiplier with error tolerance. When compared to the Vedic multiplier, boothmultiplier,andWallacemultiplier,theapproximationmultiplierismore efficientwithlesspower,delay,and area.Theproposed approximate multiplier is developed using Xilinx ISE and virtex-4 FPGA to investigate its performance. Theresults are compared to existing multipliers such as the Wallace tree multiplier, the booth multiplier, and the Vedicmultiplier.

Table1.Comparisonofproposedapproximatemultiplierwithexistingmultiplier

Attributes	**Wallace tree multiplier**	**Boothmul tiplier**	**Vedicmul tiplier**	**Proposedap proximate multiplier**
No. of sliceLUTs	96/53200	33/53200	124/53200	29/53200
Pathdelay in ns	11.954	5.400	14.739	4.828
Powerin W	13.693	7.231	14.232	7.156

The area consumed by the multiplier is determined by the number of slices LUTs occupied by the FPGA. The delay ismeasured in nanoseconds. The results reveal that the suggested approximate multiplier is more efficient than the othermultipliersintermsoflatency,area,andpower.Whencomparedtothepresentmultipliers,theutilisationofsliceLUTsisalsoquitelow.

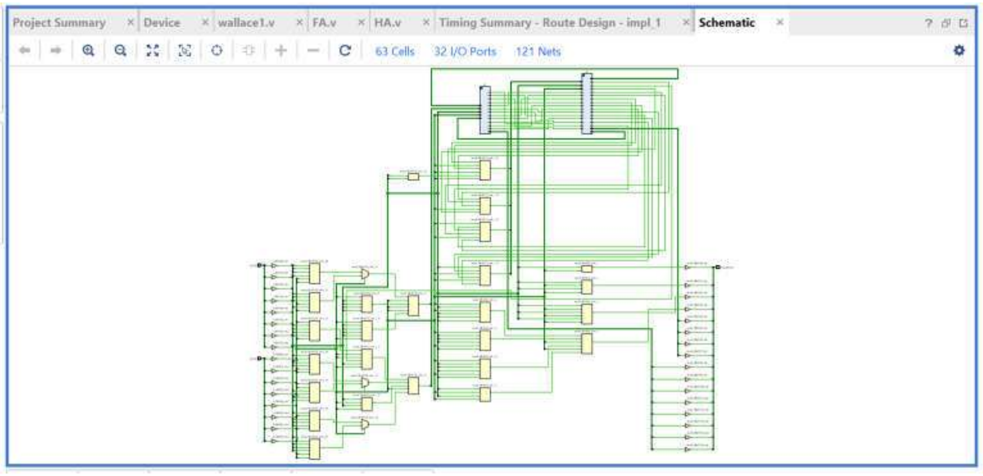


Fig18.schematicofWallacetreemultiplier

Theabove figureshowsthattheschematicdiagramofWallacetreemultiplier

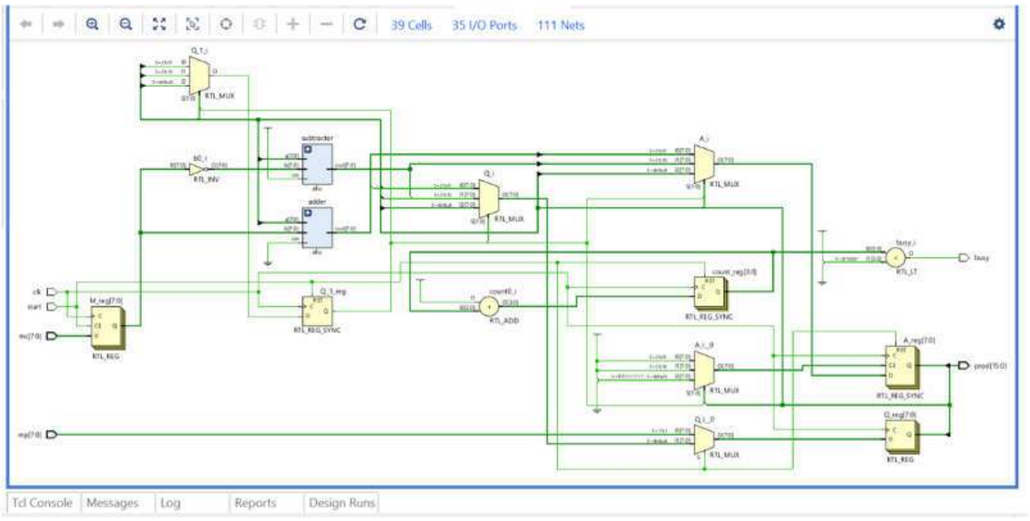


Fig19.schematicofboothmultiplier

Theabovefigureshowsthattheschematicofthebooth multiplier

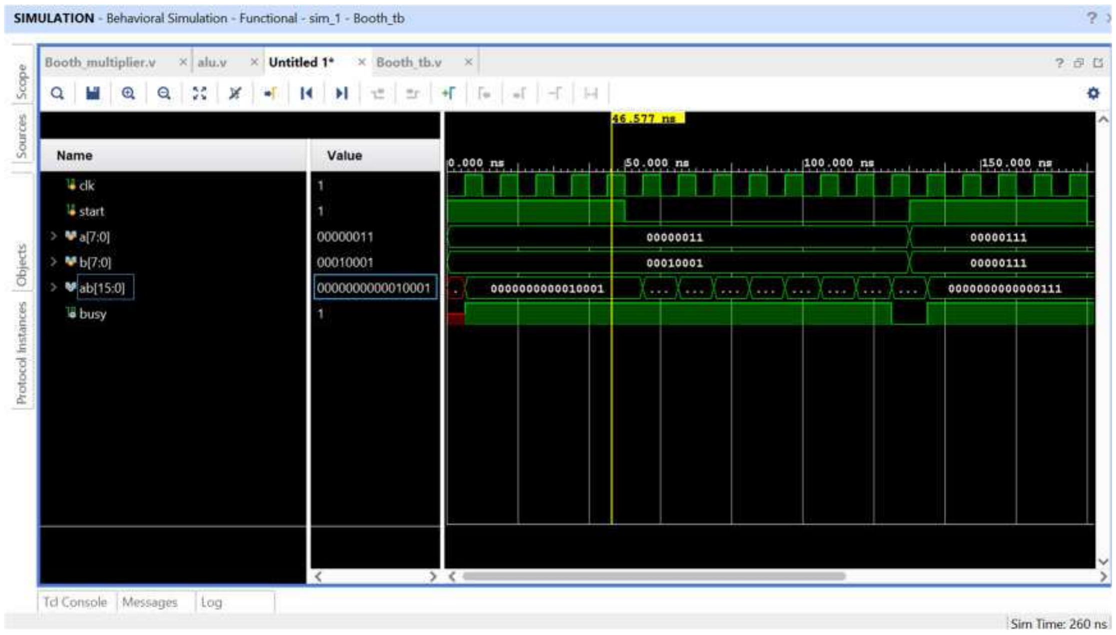


Fig20.Simulationofboothmultiplier

Theabove figureshowsthatthesimulation oftheboothmultiplier

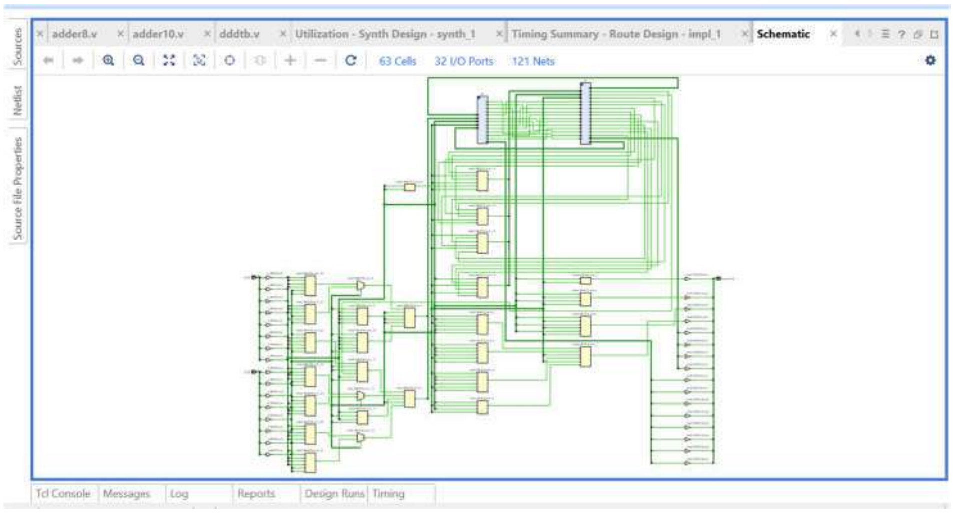
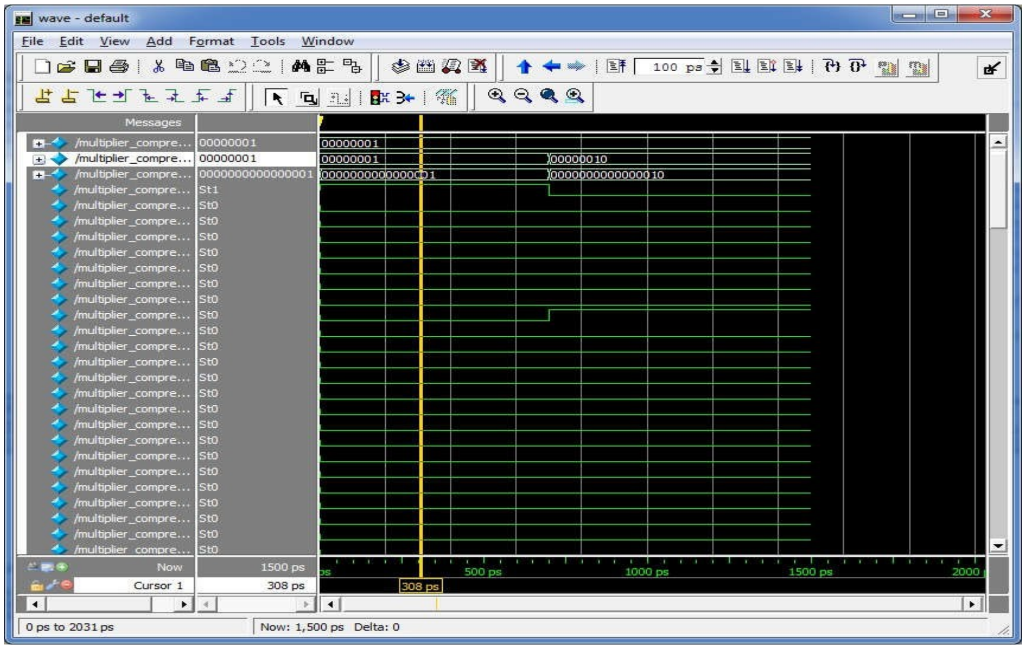


Fig21.SchematicofVedicmultiplier

TheabovefigureshowsthattheschematicoftheVedicmultiplier



```
/multiplier_compre...  00000001
/multiplier_compre...  00000001
/multiplier_compre...  0000000000000001
/multiplier_compre...  St1
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
/multiplier_compre...  St0
Now       1500 ps
Cursor 1  308 ps
0 ps to 2031 ps    Now: 1,500 ps  Delta: 0
```

Fig22.Simulationwaveformforboothapproximatemultipliers

Theabovefigureshowsthatthesimulationwave formforthebooth approximatemultiplier.

CONCLUSION

In this work, we discovered that the approximation multiplier outperforms the Wallace tree multiplier, the Vedicmultiplier, and the booth multiplier. Image processing programmes make use of multipliers. The suggested multiplieriswritteninVerilogHDL,synthesisedwithXilinx,andsimulated.Whencomparedtotheothermultipliers,thefinding sdemonstrate a significant improvement in terms of speed, power, and area. The reduction in power consumptionachieved by approximate multipliers can have significant benefits, especially in energy-constrained systems such asmobile devices or battery-powered applications. Lower power consumption leads to longer battery life, increasedefficiency, and reduced heat dissipation, which can enable the development of more compact and portable devices.Overall, approximate multipliers offer a promising approach to achieving low power consumption while maintainingacceptable levels of accuracy in various computing systems. Continued research and advancements in approximationtechniquescan furtherenhance their capabilitiesand broadentheirapplication

REFERENCES

[1] W. Liu, J. Xu, D. Wang, C. Wang, P. Montuschi and F. Lombardi, "Design and Evaluation of ApproximateLogarithmicMultipliersforLowPowerError-TolerantApplications,"in*IEEETransactionsonCircuitsandSystemsI:RegularPapers,* vol.65,no.9,pp.2856-2868,Sept.2018.
[2] K.C.Pathak,A.D.DarjiandJ.N.Sarvaiya,"LowpowerDaddamultiplierusingapproximatealmostfulladderandMajoritylogic-basedaddercompressors,"*2022IEEERegion10Symposium(TENSYMP)*,Mumbai,India,2022,pp.1-6.
[3] S.Mazahir, O.Hasan, R.HafizandM.Shafique,"ProbabilisticErrorAnalysis ofApproximateRecursiveMultipliers*,"inIEEE TransactionsonComputers*,vol.66,no. 11,pp.1982-1990,1 Nov.2017.
[4] Y. Zhao, T. Li, F. Dong, Q. Wang, W. He and J. Jiang, "A New Approximate Multiplier Design for Digital SignalProcessing,"2019 IEEE13thInternationalConferenceon ASIC(ASICON), Chongqing,China,2019, pp. 1-4.
[5] M. K. Sukla, K. Sethi and A. K. Panda, "Low-power and Area Efficient Approximate Multiplier with ReducedPartialProducts,"*2020IEEEVLSIDEVICECIRCUITANDSYSTEM(VLSIDCS),*Kolkata,India,2020,pp.181-186.

[6] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha and M. Pedram, "RoBA Multiplier: A Rounding-BasedApproximateMultiplierforHigh-SpeedyetEnergy-EfficientDigitalSignalProcessing,"*inIEEETransactionsonVeryLarge-ScaleIntegration(VLSI)Systems,* vol.25,no.2,pp.393-401,Feb.2017.

[7] I. Alouani, H. Ahangari, O. Ozturk and S. Niar, "A Novel Heterogeneous Approximate Multiplier for Low Powerand High Performance," in*IEEE Embedded SystemsLetters,*vol.10,no. 2,pp.45-48,June2018.

[8] S.KimandY.Kim,"High-performanceandenergy-efficientapproximatemultiplierforerror-tolerantapplications,"*2017 InternationalSoCDesign Conference(ISOCC),*Seoul,Korea(South),2017,pp.278-279.

[9] A. Mehta, S. Maurya, N. Sharief, B. M. Pranay, S. Jandhyala and S. Purini, "Accuracy-configurable approximatemultiplier with error detection and correction," *TENCON 2015 - 2015 IEEE Region 10 Conference*, Macao, China,2015,pp.1-4.

[10] S. Pabithra and S. Nageswari, "Analysis of Approximate Multiplier Using 15–4 Compressor for Error TolerantApplication," *2018InternationalConferenceonControl,Power,CommunicationandComputingTechnologies(ICCPCCT),*Kannur,India, 2018,pp.410-415.

[11] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra and G. D. Meo, "Comparison and Extension of Approximate 4-2 Compressors for Low-Power Approximate Multipliers," in IEEE Transactions on Circuits and Systems I: RegularPapers,vol.67,no.9,pp.3021-3034,Sept.2020.

[12] K.Shirane,T.Yamamoto,I.Taniguchi,Y.Hara-Azumi,S.YamashitaandH.Tomiyama,"MaximumError-AwareDesign of Approximate Array Multipliers," *2019 International SoC Design Conference (ISOCC)*, Jeju, Korea (South),2019,pp.73-74.

[13] T.U.SwathiKrishna,K.S.Riyas,Y.PremsonandR.Sakthivel,"15–4ApproximateCompressorBasedMultiplierforImageProcessing," *20182ndInternationalConferenceonTrendsinElectronicsandInformatics(ICOEI),*Tirunelveli,India, 2018,pp.671-675.

[14] F.-Y.Gu,I.-C.LinandJ.-W.Lin,"ALow-PowerandHigh-AccuracyApproximateMultiplierwithReconfigurableTruncation,"in*IEEE Access*,vol.10,pp.60447-60458,2022.