FPGA Based Multi Level Carry Save Adders

Yella Nirmala Devi[1], E.Rama Krishna[2], Dr.R.Rama Chandra[3]
[1]Dept of ECE
[2]HOD & Assistant Professor, Dept of ECE
[3]Principal & Professor
[1, 2, 3] Sri Krishna Devaraya Engineering College,Gooty,A.P

Abstract- *Multi operand addition, which is often found in partial product reduction of multipliers, or some combinations of addition and multiplication, is a fundamental and frequently used arithmetic operation .Though it can be realized with carry-propagate adder (CPA) trees, fast multi-operand addition usually consists of two phases, where the number of addends is compressed to 2 such as a Wallace tree and a Dadda tree, and then the final CPA generates the result of multiplication for ASIC implementation. Such trees are often constructed using 3-input 2-output counters (also called carry-save adder or full adder) and 2-input 2-output counters (half adder) as basic components.*

In this paper we prove that there is possibility to implement carry-save adders on FPGA devices with a similar hardware cost to that of carry-propagate adders, while keeping a constant computation time, in such a way that considering operands with number of bits greater or equal to 16, the speed gain is notorious, this process is similar to an ASIC-based design.

Keywords- CSTR-PID-ZN-Fuzzy-MRAM-MATLAB.

I. LITERATURE SURVEY

[1] J.-L. Beuchat and J.-M.Muller, "Automatic generation of modular mul-tipliers for fpga applications," IEEE Transactions on Computers, vol. 57, no. 12, pp. 1600–1613, December 2008.As redundant digitsystemagree toused forstable time addition, they be often at the heart of modular multipliers designed for public-key cryptography (PKC) applications. Indeed, PKC involves large operands (160 to 1,024 bits), and several researchers proposed carry-save or borrow-save algorithms. However, these number systems do not take advantage of the dedicated carry logic available in modern Field-Programmable Gate Arrays (FPGAs).

To overcome this problem, we suggest to perform modular multiplication in a high-radix carry-save number system, where a sum bit of the carry-save representation is replaced by a sum word. Two digits are then added by means of a small Carry-Ripple Adder (CRA). Furthermore, we propose an algorithm that selects the best high-radix carry-save representation for a given modulus and generates a synthesizable VHDL description of the operator.Index Terms—Modular multiplication, high-radix carry-save number system, FPGA

[2] J. Detrey, F. de Dinechin, and X. Pujol, "Return of the hardware floating-point elementary function," in Proceedingsof the 18th IEEE Symposium on Computer Arithmetic (Montpellier, France), Kornerup and Muller, Eds. Los Alamitos, CA: IEEE Computer Society Press, June 2007, pp. 161–168.

The study of specific hardware circuits for the evaluation of floating-point elementary functions was once an active research area, until it was realized that these functions were not frequent enough to justify dedicating silicon to them. Research then turned to software functions. This situation may be about to change again with the advent of reconfigurable co-processors based on field-programmable gate arrays. Such co-processors now have a capacity that allows them to accommodate double-precision floating-point computing.

Hardware operators for elementary functions targeted to such platforms have the potential to vastly outperform software functions, and will not permanently waste silicon resources. This article studies the optimization, for this target technology, of operators for the exponential and logarithm functions up to double-precision. These operators are freely available from www.ens-lyon.fr/LIP/Arenaire/.

Carry Save Adder:

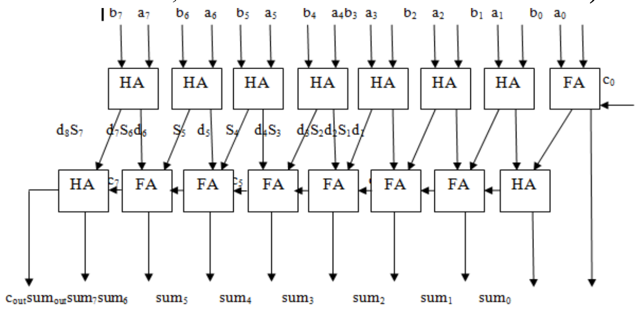
CSA is category of DA, recycled at system micro architecture calculate sum of 3 otherwise further n bit numbers binary. Diverges commencing former digital adders that outputs 2 numbers of equivalent dimensions inputs, 1 which is arrangement partial sum knobs addition alternative which arrangement carry knobs.Contemplate the sum:12345678+87654322=100000000.

By means of simple calculation, analyze left from right, "8+2=0, carry 1", "7+2+1=0, carry 1", "6+3+1=0, carry

1", besides so on completion of sum. Even though preceding digit consequence once, leading digit in anticipation of each digit calculation, passing carry commencing each digit to 1 its left. Consequently adding 2n digit information receipts time proportional n, even equipment spending might not talented performing various scheming concurrently.

Electronic expressions, by means of knobs, that mean seven n 1 bit summers at disposal, permit time proportional n permit possible carry to spread commencing unique end of number to former. Until,

1. We don't know consequence of addition.
2. We don't know consequence of the addition is larger or else smaller than a given number or not (for instance, we don't know it is +ve otherwise -ve).


b_7 a_7 b_6 a_6 b_5 a_5 b_4 a_4 b_3 a_3 b_2 a_2 b_1 a_1 b_0 a_0
HA
FA
c_0
d_8S_7 $d_7S_6d_6$ S_5 d_5 S_4 d_4S_3 $d_3S_2d_2S_1d_1$
c_7
c_5
$c_{out}sum_{out}sum_7sum_6$ sum_5 sum_4 sum_3 sum_2 sum_1 sum_0
Fig. 1 8-bit Carry Save Adder

CLA may diminish delay. In belief delay may be condensed consequently is proportional to logn, on the other hand aimed at large numbers no longer case, for the reason even as soon as CLA implemented, distances signals have travel on chip rise proportion n, as well as transmission delays rise same rate. As soon as 512 bit near 2048 bit quantity amounts essential public key cryptography, CLA is not considerably helpful.

II. DESIGN AND IMPLEMENTATION OF CARRY SAVE ADDERS ON FPGA

A make use of FPGAs near execute arithmetical circuit contain be increasing inside current existence. Inside adding in the way of their restructure capability, present FPGAs consent to elevated corresponding compute. FPGAs accomplish rate ups of two instructions of extent more than a general-purpose computer used format hematics rigorous algorithms. Therefore, these kind of strategy be more and more preferred because the goal expertise designed for a lot of applications, particularly inside arithmetical indication dispensation hardware accelerators cryptography, in addition to a large extent. Consequently, the competent completion of wide spread operator on top of FPGAs be of enormous application. The characteristic arrangement of FPGA tool surrounding substance of LEs, every one bounded through inter connection capital. In universal, all configurable elements are essentially collected of one or more than a small number of n-input search for table with flip-flops. Though, into current FPGA architectures, the collection of LEs have been greater than before by counting particular circuitry, such as devoted multipliers, obstruct RAM, with consequently taking place. Inside the author make obvious with the intention of the concentrated make use these new basics reduce the presentation GAP among FPGA with ASIC implementations. Single of these capital is the carry-chain arrangement, which be use to get better the completion of CPAs. It mostly consists of additional particular reason to contract through the take signal, with exact quick steering appearance among successive LEs, since exposed in Fig.7.1. it reserve is obtainable in the majority current FPGA devices as of low-priced ones to high-end family, in addition to it accelerate the take extend through additional than single arrange of extent compare toward its completion by common resources. not together as of the CPA completion, a lot of study have established the significance of by this reserve to accomplish design by means of improved presentation and/or fewer region supplies, and still for implement non mathematics circuit. Multi operand addition appear into a lot of algorithms, such because reproduction, filter and others. toward attain competent Implementations of this process, superfluous adders beat length second-hand. superfluous depiction reduce the count instant by defensive the extent of the Carry-propagation shackles. The majority common representations are carry-save and signed-digit . A CS adder add three statistics by an collection of Full-Adders, except lacking propagate the carry. Within this container, the FA is typically recognized because a 3:2 counter. The consequence is a CS numeral, which is composed of a sum-word and a carry-word. Therefore, the CS result is obtained without any carry propagation in the time taken by only one FA. The addition of two CS numbers require an collection of 4:2 compressors, which container be implement through two 3:2 counter. The adaptation to no redundant symbol is achieve by addition the summation and carry expression in a conservative CPA. Though, suitable to the competent completion of CPAs, the utilize of superfluous adders have typically be discarded as target FPGA tools.

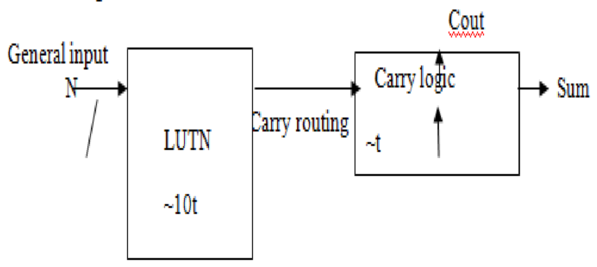


Fig.2 Carry-chain resources in modern FPGA devices.

A straight execution of a 3:2 counter typically double the region supplies of its corresponding CPA and enhanced rate is simply obvious used for extended bit widths. however, more than a few current study have established to outmoded adders be able to exist professionally map on FPGA structure, dipping region overhead and improving speed. in spite of the significant advance represent by these previous studies, the answer prospect need also (or from time to time both) the use of a complicated heuristic to create each compressor tree or a low-level drawing. The final obstruct portability, as it is extremely needy on top of the central.Structure. With in calculation, their region with rate can be enhanced, as the make use of a particular quick carry-chain is extremely incomplete.

Here, we study the competent execution of multi operand redundant compressor trees in current FPGAs by their fast carry capital. Our approach powerfully reduces holdup and they usually there no region in the clouds compare to a carry propagate adder tree. Furthermore, they can be distinct at a elevated stage on of standard CPAs. As a consequence, they are compatible with any field programmable gate array family or brand, here development in the CPA system of future FPGA families would also benefit from them. Moreover, its simple structure, it is easy to design a parametric HDL core, it allows a compressor tree synthesizing for any number of operands of any bit width. Our design presents better performance compared to previous approaches because it is easier to implement, and offers direct portability.

The relax of the document focus on top of CS illustration, as the addition toward SD illustration can be only achieve through invert convinced input and output signal beginning and toward the compressor tree, because be established within . as it is redundant to create one interior change toward the array structure, these little modification perform not considerably adapt compressor tree presentation.
here the make use of high-radix CS illustration is future, i.e., convertex tended CPAs interested in the added extras of small digit. though, this unusual illustration have significant limits; intended for illustration, spot broken up is not allowable. within the completion of a radix-4 SD adder on 6-LUT-based FPGAs is address, except carry property be not use with the region slide is unmoving extremely elevated (88 percent extra logic resources). Low-level design, i.e., use and straight configuring FPGA primitives, be use within and to drawing characteristic binary redundant compressors on 4-LUT-based FPGAs with carry resources. Together study concludes that a 4:2 compressor obtain the best presentation and produce no area in the clouds on Xilinx FPGAs. All of these studies only focus on the optimization of secluded adders, except they act not address the topic of compressor tree design. none the less, they could be used as fundamental structure block toward assemble characteristic compressor trees. Corresponding multi operand addition is addressed for 6-LUTbased FPGAs by Parandeh-Afshar and by Matsunaga etc. All of these study there compressor tree design base on top of GPCs .primary, more than a few GPC size are correctly select and characterize to resourcefully use the interior resources of the aim FPGA. next, each one study propose a diverse algorithm to construct the exact compressor tree based on a system of GPCs in such a way that an effort is complete to reduce the dangerous pathway with/or else the region of the tree. through the exemption of, the GPCs are implement by the universal logic resource alone. In common, they details significantly compact delays and a temperate enhance in area compare to ternary CPA trees. Their main drawback are that they are not applicable for 4-LUT-based FPGAs, they defer erratic results, and they necessitate the utilize of software to design each specific compressor tree.

III. CS COMPRESSORTREES ON FPGAs

here this segment, we there dissimilar approach to competently drawing CS compressor trees on FPGA devices. within accumulation, estimated delay and area analysis conducted to the general case.the generic compressor tree consider of Nop input operands with N bit width. Here assume the same bit width for input and output operands. therefore, input operands are be supposed to have before be sign or zero extensive to assurance that no run over occurs. The multi operand CS addition is provide a complete analysis of the number of leading guard bits required.

Regular CS compressor tree design

The design of multi operand carry save compressor tree to reduce the number of levels in its structure.heremost used building blocks implement for 3:2 or 4:2 compressor.In Xilinx FPGAs a 4:2 compressor is basic building it is efficiently implemented.The CS compressor tree execution requires [Nop/2]-1, 4:2 compressors (here single one eliminates two signals), where a carry-propagate tree use.

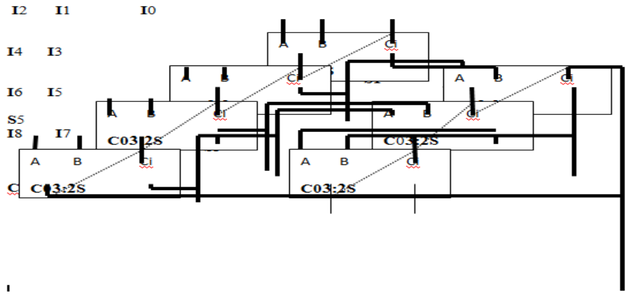


Fig.3 CS 9:2 compressor tree based on a linear array of CSAs for N-bit width.

Nop-1 CPAs (single one eliminates one signal). In the condition 4:2 compressor use almost twice the quantity of resources as CPAs, both trees basically require the same area. on top of the other hand, the speed of a compressor tree is resolute by the number of levels required. Within this case, Because single level halve the number of input signals, the critical path delay is roughly.

$$L_{4:2}=\lceil log_2(N_{0p})\rceil_{-1}$$

D $\sim L_{4:2}\cdot d_{4:2}$

Where $L_{4:2}$ is number of levels and $d_{4:2}$ is delay of a 4:2 compressor level (together with routing). Each 4:2 compressor tree structure to assume similar delay for all paths. the delay from the carry input to output and the routing to the next input is usually more than one order of magnitude quicker than the rest of the paths concerned in connecting two FAs (see Fig.7.1). The full adder carry-chain conserved possible to find fast circuits. this is the idea for structure of the 4:2 compressor obtainable for Xilinx FPGA. Here the idea of compressor are proposing different linear arrays. Here critical path reduced in the compressor treesparticular carry-chains.

Linear array structure

Here the design of single 4:2 compressor are used specialized carry resources.But the entire compressor tree structure are not be considered these resources.The propose compressor tree is similar to linear array of carry save adders.in our case two output words are given in each adder i.e sum and carry words ,

Only the carry word is connected from each next carry save adder, the Sum word are connected to the next lower level of array. Fig.7.2 shows 9:2 compressor tree posed linear structure example. Here all lines are N-bit width buses, and correctly shifted carry signals. The carry save adder compare between the regular A and B inputs and carry input is Cin, carry input and output is fast carry resources in between the dashed line representation.

The first carry save adder is exception, then here introduced the input operand Cin we can see the fig 7.2 here the entire carry-chain preserved from the input to output of compressor tree (I0 to Cf).

The first two regular inputs on each carry save adder used to add all input operands (Ii)input and output represents the fast carry resources. With the exception of the first CSA, where Ci is used to introduce an input operand, on each CSA Ci is joined to the carry output (Co) of the before CSA, as shown in Fig. 3.2. Thus, the whole carry-chain is conserved from the input to the output of the compressor tree (from I0 to Cf). Primary, the two regular inputs on each CSA are used to add all the input operands (Ii).

Input operands all introduced in the array, sum words are before generated and added in order i.e first created partial sums are added first.

In this propagation overlap between the carry chain and regular signals. Area regarding to the generic compressor tree implementation based on N bit width carry save adders it requires Nop/2 it elements (here CSA eliminates one input). Carry save adder considering to implemented using the same number of a binary carry propagated array. it is proposed linear array. binary CPA tree and 4:2 compressor tree are same hardware cots approximately. A classic view our compressor tree has Nop/2 levels for the delay relation. It is more than Wallace tree structure here it is longer critical path. But we are targeting an FPGA implementation, so we assume we temporarily no delay for carry-chain path.This assumption carry signal links be eliminated from critical path and linear array is hypothetical tree shown in fig.7.3.(where represent carry-chain in gray).

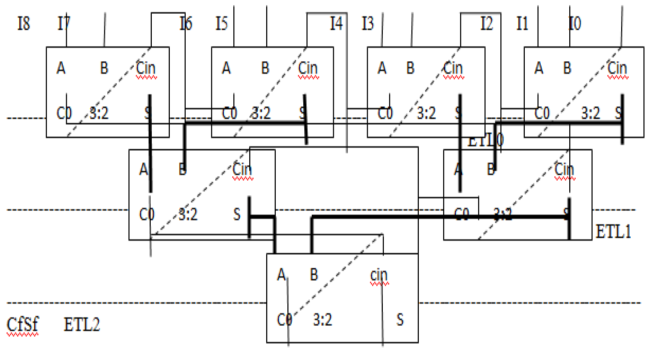


Fig.4 proposed 9:2 compressor tree in time mode of carry save.

Hypothetical tree compute the number of “effective time levels”(ETL).in the carry save adder each one considered a2:1 adder, first one except, it is 3:1 adder, the level first adders formed first bð(Nop1-1)Þ=2c bðNop1Þ=2c .

The first effective time level produces bðNop1Þ=2c partial sum words it added to second level of carry save adder (Nop last input is even), effective time level of carry save adder halves the number input to next level.

Delay is approximately L times in this tree of effective level signal.

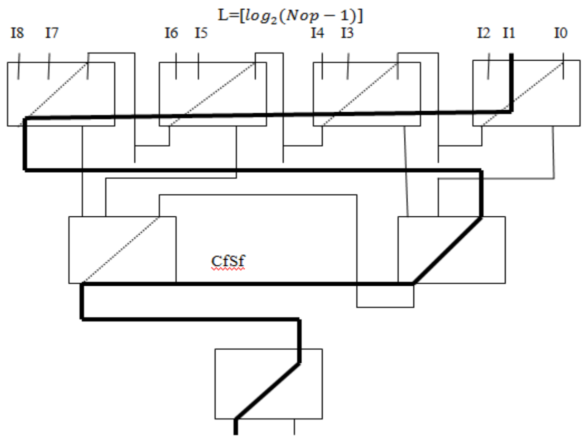


Fig.5.The proposed critical path of 9:2 compressor tree for linear array behavior.

The carry-chain delay is comparatively low,but not null.now consider the delay two global signals are d_{carry} , delay for path between the carry inputs (Ci) in the carry save adder, d_{sum} is the delay from the one general input A and B in carry save adder. One is the delay of entire carry chain occurs to each ETL (the number of carry save adders of effective time level is d_{carry},instead first carry signal generated from I1,I2,I3 in CSA and carry signal propagate to the entire carry chain output.

$$D_{low} \sim d_{sum} + (Nop-3).d_{carry}$$

Here d_{sum} **is** usuallyorder to magnitude greater thand_{carry}.It condition partially full filled for High Nop wit compressor tree.

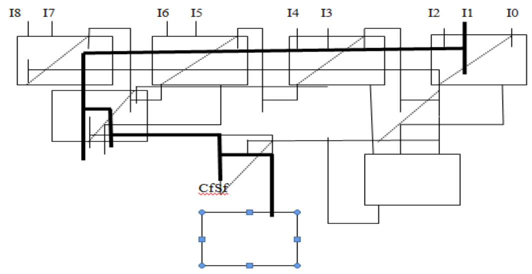


Fig.6 The proposed critical path of 9:2 compressor tree for tree behavior.

Here (i.e., in all ETLs) can only be totally fulfilled if dsum<dcarry , which is not probable on FPGAs.Other extreme situation occurs the delay of carry-chain on each ETL (d_{carry} the number of carry save adder of effective time level) is always less than delay from an ETL to next one(d_{sum}) the hypothetical tree presented in fig7.3 critical path is shown fig 7.5.

$$D_{up} \sim [(N_{op}-1)].d_{sum} + \frac{N_{op}-1}{2}.d_{carry}$$

This conditionbeextremelyrecurrentasbut the delay all throughout the carry-chain of the first ETL is less thand_{sum}, after that ETLs grab this condition. Thus, only the following condition:

$$d_{sum} > \frac{N_{op}-1}{2}.d_{carry}$$

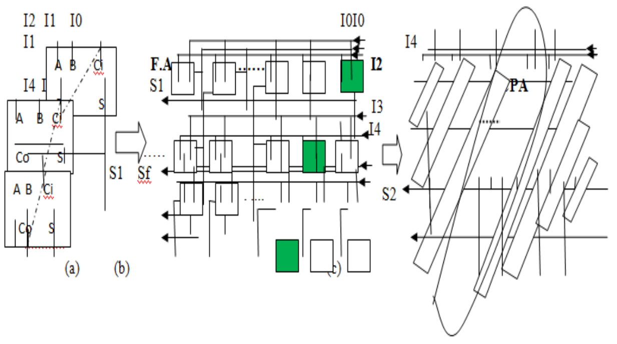


Fig.7. N-bit Translation of 5:2 linear array compressor tree into CPA array.

it satisfied, for value of Nop up to 2d_{sum} =d_{carry},linear structure of delay compressor tree very close upD_{up} .values of Nop the delay of the compressor tree is amongD_{up} and D_{down}, the compressor tree is a mix of both situation. The first CSAs form a linear arrangement awaiting the delay of the carry-chain in an ETL is lower than d sum, and then the remaining ones form a theoretical tree. Here any case, the behavior of the delay related to the number of operands has two stabilizer components: 1) a irregular logarithmic deviation due to the number of ETLs of the theoretical tree; and 2) a linear variation.

High-Level Implementation

In FPGA design a high-level explanation using HDL presents some significant **advantages**, follow as different families portability, Simplification is easier, During the production is lower error. Disadvantages are the final implementation control is lost. Unexpected results is produce. From the higher level circuit is achieve some times only efficient implementation. the design is anchored to specific FPGA inner structure. It is the case of classic 4:2 compressor proposed in the section 3.1 for 4-LUT based FPGA families.

Improvement for Ternary Adders

To develop the presentation of multioper and accumulation, latest FPGA families, such as Virtex-5/7 or Stratix-II/V, it capably implement ternary additionðAþBþC¼DÞ.A simple 2-input is showing similar speed in ternary adder requires the same amount of income. as single ternary adder eliminates two operands, compressor tree is the number adders requiresdðNop1Þ=2e, which is binary case is almost half the amount needed. On the other hand, the number of levels is

$$L_{3:1}=[\lceil log_3(N_{op})\rceil]$$

It is faster than binary adders, it preferred to implement parallel multi operand addition it target devices. here now present new resources it is how to compressor the linear array design is adapted to take advantage. the integration of initial 3:2 carry save adder on based in ternary adder construction too with binary carry propagate array in same logic elements. A the ternary adder 1-bit element are three operand input signals, two carry signals and one sum output having two input and two outputs. one carry signal cA,is related to carry save adder,its have limited delayi.e the carry signal generation not depend to before value of carry. other cB,is related to carry propagate array.it form a carry-chain

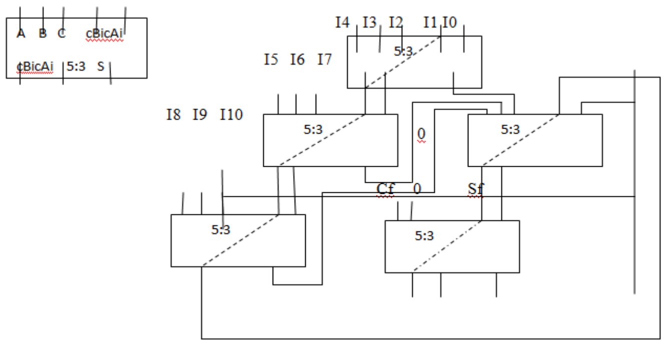


Fig 8. CS 11:2 compressor tree based on a linear array of 5:3compressors.

$$N_{add} = [\frac{N_{op-1}}{2}]$$

Similar carry ripple adder converted to carry save adder.the ternary becomes a 5:3 compressor array here carry signals are disconnected.The 5:3 compressor array constructed in an efficient compressor tree as shown fig.7.7. In before approaches first exception in compressor array. here 2 operands are additional ,two carry input are connected to carry outputs of previous adder. There partial sum words are added to same order to produced next all input operands. The 5:3 compressor array of first five operands it requires to implement a compressor tree because two operands are eliminated by each compressor array.in the ternary adders of FPGA resources in each N-bit width 5:3 compressor array. The delay for the extreme cases approximately are

$$D_{low} \approx d_{5:3}+([\frac{N_{op}}{2}]-1).d_{cB}$$
$$D_{up} \approx ([log_3(N_{op})]+1).d_{5:3}+([\frac{N_{op}}{2}]-1).d_{cB}$$

here d_{cB} delay matching to path between the cB inputs of two successive 5:3 compressor array and $d_{5:3}$ is delay after one ETL to the next, i.e., the general input of compressor array is joined to sum wordof before compressor array. N add-bit width in terms of structurally required to implement it. One time again, the most noteworthy sum-bit of single ternary adder comprise sum-word of compressor tree, while the last cB out is the last carry-word. apart from for the end adders, single ternary adder sums one bit each operand and partial sum, unreliable the bit heaviness depending on its relation position. stipulation adders and bits are numbered as for the binary case, the h bit of the ternary adder jadds the bitjþi,NAddþ1of the operands3iþ2, 3iþ3, and3iþ4. Note that the operands 0 and 1 are linked to the two carry inputs of each ternary adder and negative bit positionbe set to nil

If linear arrangement is professionally mapped on any FPGA machine suitable to put into practice ternary adders. These plans are the newer FPGAs based on 6-LUT or 8-LUT, such as Stratix-II/V for Altera or Virtex-5/7 for Xilinx. The high-levelaccount of this compressor tree could be used by any software tool able of target ternary adders as likely basic rudiments. inside other cases, a broad ternary adder should be before defined.

7.2.5Pipelining :

The major advantages of CS compressor trees are that they are very appropriate for pipelining. An M-stage pipeline compressor tree of Llevels of compressors is implement just by introduce one level of register for every L=M levels of compressors. though, the futuredesign are distinctby an array of CPAs, which complicate the foreword of register in the linear array compressor tree. Though, the proposed move toward is motionless simple to be relevant in pipeline design. The pipeline compressor tree is designed by utilize smaller linear array compressor trees as basic building block, which are register in the input or output. For example, a two-stage 18:2 compressor tree is built by using three 6:2 linear array compressor trees. The best size for the linear array block depends on the number of operands (Nop) and the number of stages (S) required for our design, but the best size could befound using

$$X=[2^{\sqrt[s]{\frac{N_{op}}{2}}}]$$

inside the case, to the time supplies are known in its place of the number of stages, the graph shown in Section 4 couldbe used to find the size that best meet these supplies.

IV. IMPLEMENTATION RESULTS AND COMPARISION

FPGA Families results on with Support forBinary CPAs

Used to sake of unfussiness, for two FPGA family tested(Spartan-3A and Virtex-4), just the consequences matching to the Virtex-4 obtainable. Ternary adders are not efficiently implemented, consequently, here only tested ones on binary adders. Now denote as carry propagate array is classic tree, the proposed OUR array linear array construction base on 3:2 carry save adder and4:2 tree the typical tree arrangement based on compressors. concerning the area, Fig. 7.1show the quantity of LUTs necessary by the dissimilar compressor tree arrangement when varying Nop from 4 to 128 operands, for 16 and 64 bit widths. through the exception of 4 and 5operand compressor trees, we think disconnectedly, three compressor trees are very similar and linearly varies number of operands and bit widths are used in area, because predictable. specially, CPA tree and OUR array be area almost the same, while the 4:2 compressor tree require a little additional area (up to 6 percent for a 16-bit width and up to 2 percent for a 64-bit width), due to the conclusion of border bits on the 4:2 CA. Let us now think the cases of four and five operands.

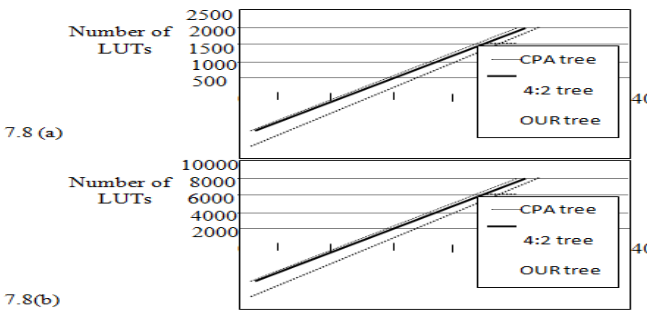
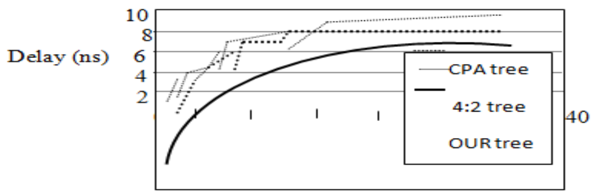


fig 9: Area (LUTs) of the different compressor tree approaches

The CPAs concerned in the completion of OUR array are only 2 and 3 bit width for all operand sizes. Known this small size, the mixture tool equipment these CPAs by completely using LUTs, and not the particular carry-chain, because this produces faster circuits. As a result, there is an add to in area for these exacting cases (as shown in Fig. 7.1), which might be**Fig.9 Area (LUTs) of the different compressor tree approaches implemented on Virtex-4 when varying the number of input operands (Nop for (a) 16-bit width and (b) 64-bit**though, in it case, the accelerate achieve is almost independent of the figure of bits N. We be supposed to also letter so as to while the bit width Nis extremely low compare to Nop, the utmost delay OUR array is incomplete suitable to result of end carry propagate arrays. For this reason, in Fig. 7.3, the delay OUR array for extra than 64 elements is lesser for 16-bit width (see Fig. 7.3a) than for 64-bit width (see Fig. 7.3bb). Table 1 present the limits and the standard of the speed ups obtain via use OUR arrayor4:2treeinstead of CPA tree, intended ford is similar range of Nop. The OUR array advance be obviously higher to the classic4:2 tree, and it can attain powerfully compact delay, even for small bit widths.

FPGA Families results with Support for Ternary CPAs

We contain synthesize every one the compressor tree structure, unreliable Nop starting 4 to 128 elements for N equal to 16 and 64 bits.



(b) Number of operands (Nop)

Fig 10. Delay (ns) of different compressor tree approach development on Virtex-4 while unstable the digit of input operands (Nop) for (a) 16-bit width and (b) 64-bit width.

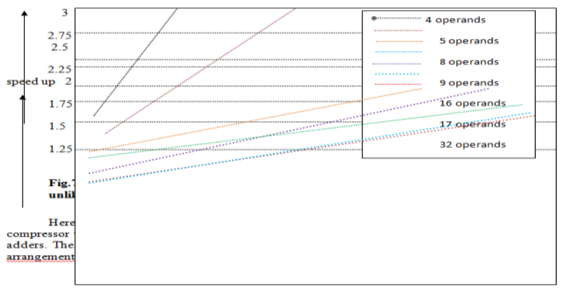
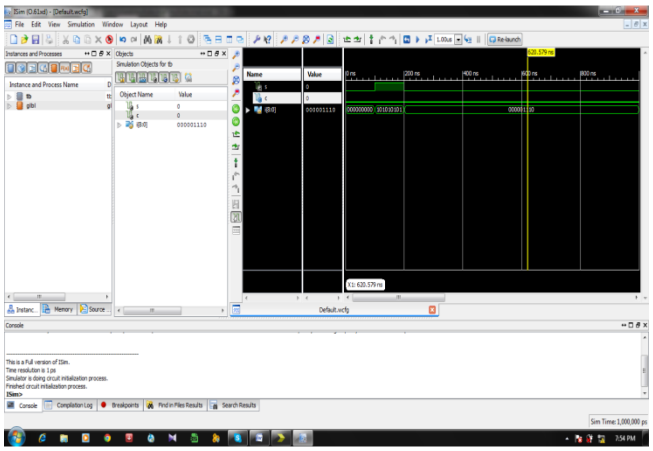


fig 10:Bounds and Average of the Speedup Achieved on Virtex-4 by Using OUR Array and 4:2 Tree Instead of CPA Tree

Nop	4-16	17-32	33-64	65-128
N	OUR array vs tree (main/mean/max)			
16	1.19/1.46/2.29	1.12/1.21/1.35	1.06/1.16/1.27	1.04/1.11/1.20
32	1.33/1.66/2.80/	1.23/1.33/1.49	1.14/1.25/1.38	1.05/1.15/1.26
64	1.61/2.08/3.81	1.44/1.56/1.76	1.31/1.43/1.59	1.14/1.27/1.43
128	2.17/2.92/5.85	1.87/2.03/2.28	1.65/1.81/2.01	1.36/1.55/1.76
	4:2 tree vs CPA tree n(min/mean/max)			
16	-0.97/1.06/1.52	0.90/0.91/0.91	0.86/0.87/0.87	0.83/0.84/0.84
32	1.08/1.21/1.86	0.99/0.99/1	0.93/0.93/0.94	0.89/0.90/0.90
64	1.31/1.51/2.54	1.16/1.17/1.17	1.16/ 1.17/1.17	1.01/1.01/1.01
128	1.77/2.11/3.90	1.50/1.52/1.52	1.32/1.36/1.36	1.24/1.24/1.25

V. RESULTS AND DISCUSSION



```
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Fig.11 Simulation Result

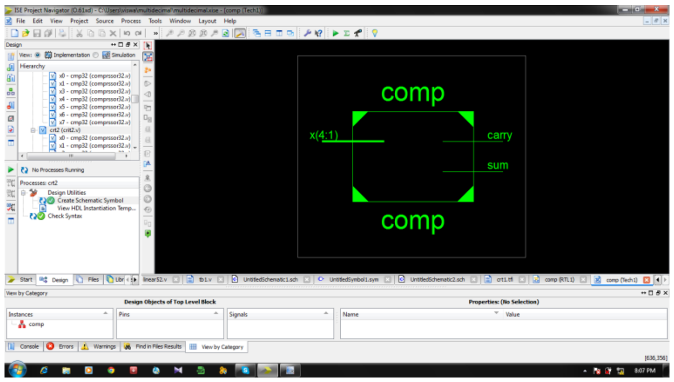


Fig.12 RTL schematic RTL schematic represents the block diagram representation of the hardware internal diagram.

VI. CONCLUSIONS

Efficiently implementing CS compressor trees on FPGA, here area and speed, be complete probable with by the particular carry-chains that devices into a original method. Related near it happen while by means of ASIC technology, the future CS linear array compressor trees guide to ward noticeable improvement into speed compare toward CPA approach and within universal, by means of no extra hardware price. In addition, the proposed complex description of CSA arrays base resting on CPAs facilitate accessibility and portability, even inside relative to prospect FPGA architectures, since CPAs determination probably stay a key part within the next generation of FPGA. We have compare our architectures, development on special FPGA families, near several designs and have provide a qualitative and quantitative learn of the benefits of our proposals.

VII. FUTURE SCOPE

Prefix adder architectures capable of three – operand addition for cell based design and their synthesis have been designed and investigates in this thesis. Binary adders capable of constant addition have also been presented and their performance investigated.

The design is possible due to the generation of a new set of intermediate outputs called “flag” bits. Adders with the possibility of having the third operand as a constant or a variable binary number. The hardware will be optimized by gate sizing in order to achieve better performance results.

REFERENCES

[1] J.-L. Beuchat and J.-M. Muller, “Automatic generation of modular mul-tipliers for fpga applications,” IEEE Transactions on Computers, vol. 57, no. 12, pp. 1600–1613, December 2008.

[2] J. Detrey, F. de Dinechin, and X. Pujol, “Returnof the hardware floating-pointelementary function,” inProceedingsof the 18th IEEE Symposium on Computer

[3] Arithmetic (Montpellier, France), Kornerup and Muller,Eds. Los Alamitos, CA: IEEE Computer Society Press,June 2007, pp. 161–168.

[4] P.Siva Nagendra Reddy and A.G.Murali Krishna “Implementation of RISC Processor for Convolution Applications” in International journal of Computer Trends and Technology, Volume 4, issue 6-2013 ISSN: 2231-2803(IJCTT).

[5] H. Eberle, G. N., S. Shantz, V. G upta, L. Rarick, andS. Sundaram, “A public-key cryptographic processor forRSA and ECC,” in Proceedings of the International

[6] Conference on Application-Specific Systems,Architectures and Processors (ASAP2004), September2004.

[7] H. R. Ismail, R.C., “High performance complexnumber multiplier using booth-wallace algorithm,” in IEEEInternational Conference on Semiconductor ElectronicsICSE, November 2006.

[8] K. Manochehri and S. Pourmozafari, “Modified radix-2 montgomery modular multiplication to make it fasterand simpler,” in IEEE International Conference on

[9] Information Technology: Coding and Computing, ITCC2005 , April 2005.

[10] M.D.ErcegovacandT.Lang, Digital Arithmetic.Morgan Kaufmann.

[11] R.NareshNaik , P.Siva Nagendra Reddy and K. Madan Mohan “Design of Vedic Multiplier for Digital Signal Processing Applications” in International Journal of Engineering Trends and Technology, volume 4 issue 7-2013 ISSN: 2231-5381(IJETT).

[12] P.Siva Nagendra Reddy et all.."Design and Implementation of FPGA based 64-bit MAC Unit using VEDIC Multiplier and Reversible Logic Gates" Indian Journal of Science and Technology, Vol 10(3),DOI:10.17485/ijst/2017/v10i3/109413, January 2017