International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Design and Reducing of power and area for Parallel prefix adders

M. Parameshwara Rao M.Tech Student in Dept of ECE
Sri mittapalli College of Engineering
parimi.mallam@gmail.com
M. Nageswara Rao Assoc. Professor in department of ECE
Sri Mittapalli College of Engineering.

ABSTRACT: The hardware implementation of binary addition is a fundamental architectural component in many processors, such as microprocessors, digital signal processors, mobile devices and other hardware applications. Currently, parallel prefix adders (PPA) are considered effective combinational circuits for performing the binary addition of two multi-bit numbers. These adders are widely used in arithmetic-logic units, which are parts of modern processors, such as microprocessors, digital signal processors, etc. This paper deals with Kogge-Stone adder, which is one of the fastest PPA. When performing the schematic implementation, this adder has a large hardware complexity. Therefore, in this work for reducing its hardware complexity the scheme of modified PPA has been developed. The performance parameters considered for the comparative analysis of the presented adders.
Project will be developed using verilog HDL. Xilinx ISE tool is used to perform the Simulation and Synthesis.

1.INTRODUCTION

Albeit parallel counts are the predominant in many machines, they are not reasonable for business, banking, and business applications because of the inadmissible inaccurate decimal to paired transformation mistakes they produce. A genuine model demonstrates the outrageous impact of these off-base approximations, where it expressed that if a correspondence organization approximates a 5% deals expense on a thing, (for example, a $0.70 phone call), the yearly misfortune is over than a $5 million.
In Processors (DSP) and chip information way units, adder is a significant component. All things considered, broad research keeps on being centered around improving the power-postpone execution of the adder. In VLSI executions, parallel-prefix adders are known to have the best execution. Reconfigurable rationale, for example, Field Programmable Gate Arrays (FPGAs) has been picking up in ubiquity as of late in light of the fact that it offers improved execution as far as speed and control over DSP-based and chip based answers for some, handy structures including versatile DSP and broadcast communications applications and a critical decrease being developed time and cost over Application Specific Integrated Circuit (ASIC) plans. The power bit of leeway is particularly significant with the developing notoriety of versatile and convenient hardware, which utilize DSP capacities. Be that as it may, as a result of the structure of the configurable rationale and steering assets in FPGAs, parallel-prefix adders will have an unexpected exhibition in comparison to VLSI executions. Specifically, most present day FPGAs utilize a quick convey chain which streamlines the convey way for the basic Ripple Carry Adder (RCA).

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Programming imitating of decimal math, as an answer for the partial estimate issue, isn't quick enough. Contrasted with equipment speeds, the exhibition of existing decimal number-crunching programming libraries is exceptionally poor. Programming imitating is slower than an equipment execution by 100 to multiple times. As of now, decimal math is executed utilizing programming while double number-crunching is generally actualized by the equipment, though.

So as to guard the developing advancement of the decimal math, proficient decimal calculations must be examined. Decimal digit adders and decimal digit multipliers are key segments of any decimal equipment to help decimal number juggling applications. Subsequently, this work centers around conveying effective BCD digit units to be utilized in superior decimal equipment quickening agents.

Two fundamental commitments of this work can be featured: proposing two new BCD digit adders and proposing one new BCD digit multiplier. These structures are portrayed and recreated utilizing VHDL equipment depiction language. They are altogether actualized on a FPGA and contrasted and existing originators.

2.LITERATURE SURVEY

Kogge P, Stone H proposed A parallel calculation for the productive arrangement of a general class Recurrence relations . A mth-request repeat issue is characterized as the calculation of the arrangement x1, x2, ..., XN, where xi = fi(xi-1, ..., xi-m) for some capacity fi. This paper utilizes a system called recursive multiplying in a calculation for illuminating a huge class of repeat issues on parallel PCs, for example, the Iliac IV. Recursive multiplying includes the part of the calculation of a capacity into two similarly complex subfunctions whose assessment can be performed at the same time in two separate processors. Progressive part of every one of these subfunctions spreads the calculation over more processors. This calculation can be connected to any repeat condition of the structure xi = f(bi, g(ai, xi-1)) where f and g are capacities that fulfill certain distributive and affiliated like properties. In spite of the fact that this repeat is first request, all straight mth-request repeat conditions can be thrown into this structure. Reasonable applications incorporate direct repeat conditions, polynomial assessment, a few nonlinear issues, the assurance of the greatest or least of N numbers, and the arrangement of tridiagonal straight conditions. The subsequent calculation registers the whole arrangement x1, ..., xN in time corresponding to [log2 N] on a PC with N-overlap parallelism. On a sequential PC, calculation time is relative to N.

D. Harris proposed A Taxonomy of Parallel Prefix Networks Parallel prefix systems are broadly utilized in elite adders. Systems in the writing speak to tradeoffs between number of rationale levels, fanout, and wiring tracks. This paper shows a three-dimensional scientific categorization that not just depicts the tradeoffs in existing parallel prefix organizes yet in addition focuses to a group of new systems. Adders utilizing these systems are thought about utilizing the strategy for sensible exertion. The new engineering is aggressive in inertness and region for certain advances.

Alvaro Vázquez, Florent de Dinechin (Multi-operand Decimal Adder Trees for FPGAs) the innovative work of equipment plans for decimal number juggling is right now going under an exceptional action. For most part, the techniques proposed to actualize fixed and gliding point tasks are expected for ASIC structures. In this

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

manner, an immediate mapping or adjustment of these methods into a FPGA could be a long way from an ideal arrangement. Just a couple of studies have considered new strategies increasingly reasonable for FPGA usage. A fundamental activity that has not gotten enough consideration in this setting is multi-operand BCD expansion.

Performance of Parallel Prefix Adders Implemented with FPGA technology BY K. Vitoroulis and A. J. Al-Khalili

Parallel Prefix Adders have been established as the most efficient circuits for binary addition. Their regular structure and fast performance makes them particularly attractive for VLSI implementation. The classical parallel prefix adder structures that have been proposed over the years optimize for logic depth, area, fan-out and interconnect count of the logic circuits. This paper investigates the performance of parallel prefix adders implemented with FPGA technology. We report on the area requirements and critical path delay for a variety of classical parallel prefix adder structures.

3.EXISTING PARALLEL PREFIX ADDERS

Twofold expansion is the most central and regularly utilized number-crunching task. A great deal of work on adder configuration has been done as such far and numerous structures have been proposed. At the point when high task speed is required, tree structures like parallel-prefix adders are utilized.

The Parallel Prefix expansion is done in three stages, which is appeared in fig1. The essential produce and engender sign are utilized to create the convey contribution for every adder. Two distinct administrators dark and dim are utilized here.

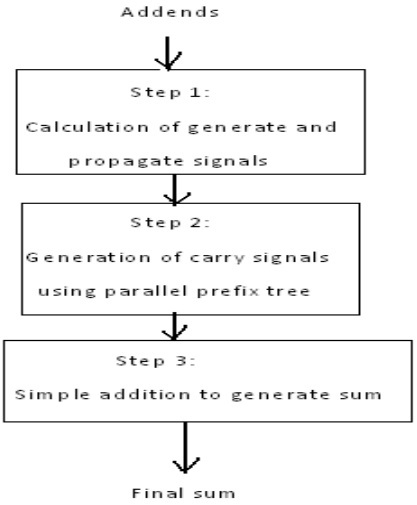


Fig 1. Addition procedure using Parallel Prefix tree structures

In every bit (i) of the two operand block, the two input signals (ai and bi) are added to the corresponding carry-in signal (carryi) to produce sum output (sumi) The equation to produce the sum output is:

Sumi = ai ^ bi ^ carryi (1)

Computation of the carry-in signals at every bit is the most critical and time – consuming operation. In the carry- look ahead scheme of adders (CLA), the focus is to design the carry-in signals for an individual bit additions. This is achieved by generating two signals, the generate (gi) and propagate (pi) using the equations:

Gi = ai ^ bi (2)

Pi = ai ^ bi (3)

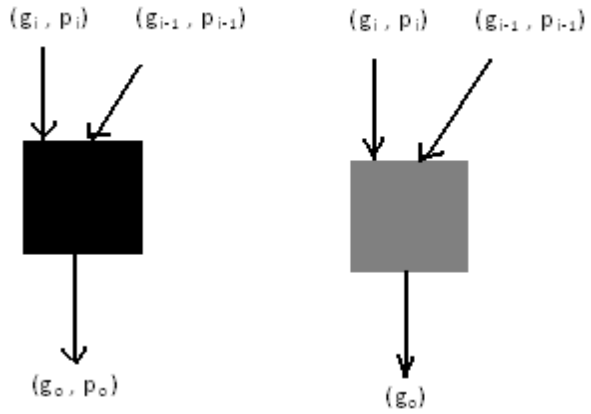
The carry in signal for any adder block is calculated by using the formula

Ci+1 = gi V (pi) (4)

Where ci must be expanded to calculate ci+1 at any level of addition

Parallel Prefix adders compute carry-in at each level of addition by combining generate and propagate signals in a different manner. Two operators namely *black* and *gray* are used in parallel prefix trees are shown in fig 2(a), fig 2(b) respectively.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362



(a) black operator (b) gray operator

Fig 2 Operators used in Parallel Prefix trees

The black operator receives two sets of generate and propagate signals (gi , pi),(gi-1 ,pi-1), computes one set of generate and propagate signals (go , po) by the following equations:

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

Po = pi ^ pi-1 (6)

The gray operator receives two sets of generate and propagate signals (gi, pi),(gi-1 ,pi-1), computes only one generate signal with the same equation as in equation (5).

It is readily apparent that a key advantage of the tree-structured adder is that the critical path due to the carry delay is on the order of *log 2N* for an N-bit wide adder. The arrangement of the prefix network gives rise to various families of adders. For a discussion of the various carry-tree structures, see [1, 3]. For this study, the focus is on the Kogge-Stone adder , known for having minimal logic depth and fanout. Here we designate BC as the black cell which generates the ordered pair in equation (1); the gray cell (GC) generates the left signal only, following . The interconnect area is known to be high, but for an FPGA with large routing overhead to begin with, this is not as important as in a VLSI implementation. The regularity of the Kogge -Stone prefix network has built in redundancy which has implications for fault-tolerant designs. The sparse Kogge -Stone adder, shown in Fig 1(b), is also studied. This hybrid design completes the summation process with a 4 bit RCA allowing the carry prefix network to be simplified

This adder computes *gi* and *hi* signals for the preprocessing stage. Then at the first level (l = 1) of prefix tree, *Gi*:*k* and *Hi*:*k* signals of 2-bit are computed within the same time. At the second level (l = 2) of prefix tree, *Gi*:*k* and *Hi*:*k* of 4-bit are calculated by using the result of 2-bit at level 1. Therefore, the actual carry-out value of the 4th bit would be available while the calculations at level 2 are being computed. At the third level (l = 3) of prefix tree, the carry-out of the 8^{th} bit is computed by using the 4th bit carry result. The same method adopted at level 3 is applied to get carry-out values of the 16th bit in fourth level (l = 4) and etc. All other carries of bit are also computed in parallel. Finally, at the final processing stage the sums are computed from these final carry-out signals of the prefix tree.

In the prefix tree the number of levels corresponds to $l = \log_2 n$ and the number of schematic nodes (white cell + gray cell) will be ($k = [n(\log_2 n) - n + 1]$). *Quine*-complexity *QK.S* and the number of logic gates *CK.S* in the *Kogge-Stone* adder are given by the following equation:

$CK.S = 3n(\log_2 n) - n + 4$ (1.7)

$QK.S = 6n(\log_2 n) - 2n + 8$ (1.8)

Figure 1 shows the scheme of a 32-bit *Kogge-Stone* adder with increasing the bit width of input operands more than 16 bits.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

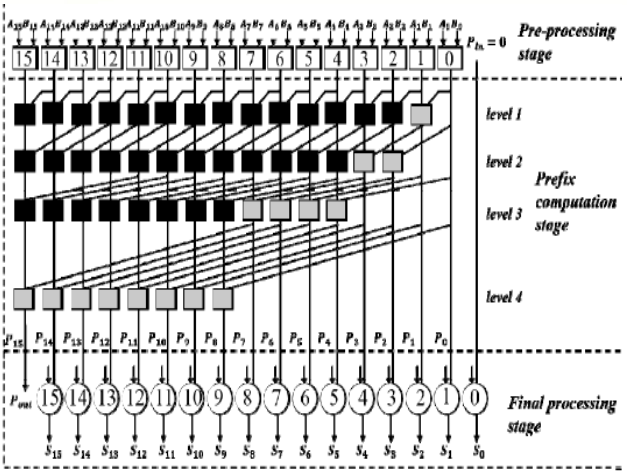


Fig3 16 bit Kogge-Stone adder

4.PROPOSED PARALLEL PREFIX ADDERS

It is readily apparent that a key advantage of the tree-structured adder is that the critical path due to the carry delay is on the order of *log 2N* for an N-bit wide adder. The arrangement of the prefix network gives rise to various families of adders. For a discussion of the various carry-tree structures, see [1, 3]. For this study, the focus is on the Kogge-Stone adder , known for having minimal logic depth and fanout. Here we designate BC as the black cell which generates the ordered pair in equation (1); the gray cell (GC) generates the left signal only, following . The interconnect area is known to be high, but for an FPGA with large routing overhead to begin with, this is not as important as in a VLSI implementation. The regularity of the Kogge -Stone prefix network has built in redundancy which has implications for fault-tolerant designs. The sparse Kogge -Stone adder, shown in Fig 1(b), is also studied. This hybrid design completes the summation process with a 4 bit RCA allowing the carry prefix network to be simplified

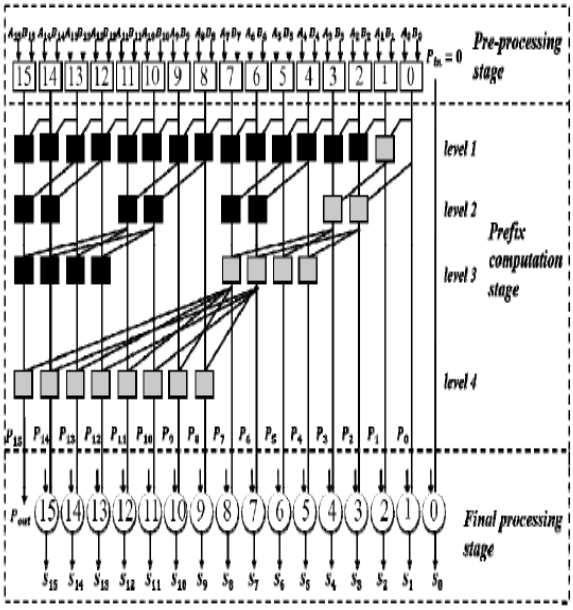


Fig4.Modified 16-bit Kogge-Stone adder

The construction of the first level of the prefix tree of this adder is similar to the construction of Kogge-Stone adder. The main structural difference begins from the second level of the prefix tree. At the second level of the prefix tree, the groups of two schematic nodes are formed, at the 3rd level – groups compose four schematic nodes and at the 4th level – groups including 8 schematic nodes, etc. This adder first computes gi and hi signals for the first stage. Then at the first level of prefix tree, Gi:k and Pi:k signals of 2-bit are computed at the same time, and then, it
computes Gi:k and Pi:k signals for pairs of columns, then for blocks of 4, then for blocks of 8, then 16, and so on until the final Gi:k signal for every column is known. Finally, at the last stage this adder computes the sums together with the generated signals obtained from the previous prefix computation stage. The number of levels of the prefix tree corresponds to (log2 n) and the number of schematic nodes

Considering the structure of the Generate-Propagate (GP) squares (i.e., the

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

BC and GC cells), we had the option to build up the accompanying plan, by considering the accompanying subset of info esteems to the GP squares.

On the off chance that we subjectively allocate the (g, p) requested sets the qualities (1, 0) = True and (0, 1) = False, at that point the table is independent and structures an OR truth table. Besides, on the off chance that the two contributions to the GP square are False, at that point the yield is False; alternately, on the off chance that the two sources of info are True, at that point the yield is True. Consequently, an information design that shifts back and forth between creating the (g, p) sets of (1, 0) and (0, 1) will compel its GP pair square to exchange states. In like manner, it is effectively observed that the GP squares being nourished by its forerunners will likewise interchange states. Accordingly, this plan will guarantee that a more terrible case postpone will be produced in the parallel prefix arrange since each square will be dynamic. So as to guarantee this plan works, the parallel prefix adders were orchestrated with the "Keep Hierarchy" structure setting turned on (something else, the FPGA compiler endeavors to redesign the rationale doled out to each LUT). With this alternative turned on, it guarantees that every GP square is mapped to one LUT, protecting the fundamental parallel prefix structure, and guaranteeing that this test methodology is powerful for deciding the basic deferral. The structures were additionally integrated for speed instead of region advancement.

(g_L, p_L)	(g_R, p_R)	$(g_L + p_L g_R, p_L p_R)$
(0,1)	(0,1)	(0,1)
(0,1)	(1,0)	(1,0)
(1,0)	(0,1)	(1,0)
(1,0)	(1,0)	(1,0)

The adders were tried with a Tektronix TLA7012 Logic Analyzer. The rationale analyzer is furnished with the 7BB4 module that gives a planning goals of 20 ps under the MagniVu setting. This permits direct estimation of the adder delays. The Spartan 3E advancement board is outfitted with a delicate touch-landing cushion which permits low capacitance association legitimately to the rationale analyzer.

5.RESULTS

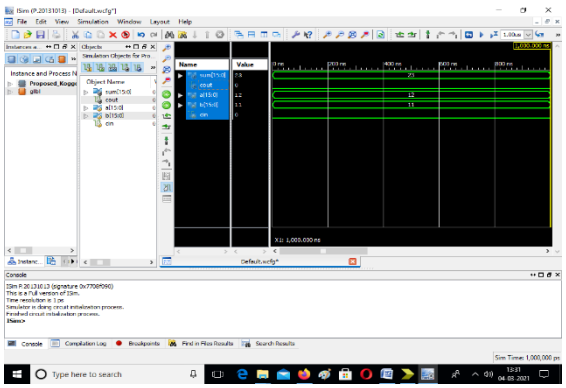


Fig.5 Output wave form

Fig 5 show the simulation result of proposed adder a, b, cin as the input and s and cout as the output

6. COMPRESSION TABLE

	EXISTING ADDER	PROPOSED ADDER
AREA	94	71
POWER	0.313w	0.311w
DELAY	12.449ns	13.335ns

7.CONCLUSION & FUTURE SCOPE

In this article, the accompanying assignments have been illuminated: examination of the point of view design for developing different multi-bit PPA plans;

induction of equations for assessing the equipment multifaceted nature of multi-bit PPA; schematic usage of the standard 16-bit and 32-bit Kogge-Stone adders and schematics usage of 16-bit what's more, 32-bit changed parallel prefix adders. At that point, a similar examination of parameters and reenactment aftereffects of the exhibited adders have been completed. Thus, looks into have appeared, that the changed parallel prefix adder proposed in the work has a bit of leeway as far as equipment multifaceted nature in correlation with the known structure of Kogge- Stone adder. Moreover, regarding speed the proposed parallel prefix adder has the favorable position over gathering prefix and convey lookahead adders, and renowned as parallel prefix adders Sklansky and Brent-Kung.

REFERENCES

[1]. K. Vitoroulis and A. J. Al-Khalili, "Performance of Parallel Prefix Adders Implemented with FPGA technology," IEEE Northeast Workshop on Circuits and Systems, pp. 498-501, Aug. 2007.

[2]. D. Gizopoulos, M. Psarakis, A. Paschalis, and Y. Zorian, "Easily Testable Cellular Carry Lookahead Adders," Journal of Electronic Testing: Theory and Applications 19, 285-298, 2003.

[3]. S. Xing and W. W. H. Yu, "FPGA Adders: Performance Evaluation and Optimal Design," IEEE Design & Test of Computers, vol. 15, no. 1, pp. 24-29, Jan. 1998.

[4]. M. Bečvář and P. Štukjunger, "Fixed-Point Arithmetic in FPGA," Acta Polytechnica, vol. 45, no. 2, pp. 67-72, 2005.

[5]. P. M. Kogge and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," IEEE Trans. on Computers, Vol. C-22, No 8, August 1973.

[6]. P. Ndai, S. Lu, D. Somesekhar, and K. Roy, "Fine-Grained Redundancy in Adders," Int. Symp. on Quality Electronic Design, pp. 317-321, March 2007.

[7]. T. Lynch and E. E. Swartzlander, "A Spanning Tree Carry Lookahead Adder," IEEE Trans. on Computers, vol. 41, no. 8, pp. 931-939, Aug. 1992.

[8]. N. H. E. Weste and D. Harris, CMOS VLSI Design, 4th edition, Pearson–Addison-Wesley, 2011.

[9]. R. P. Brent and H. T. Kung, "A regular layout for parallel adders," IEEE Trans. Comput., vol. C-31, pp. 260-264, 1982.

[10]. D. Harris, "A Taxonomy of Parallel Prefix Networks," in Proc. 37th Asilomar Conf. Signals Systems and Computers, pp. 2213–7, 2003.