International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 8th Dec 2017. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-6&issue=ISSUE-12

Title: **DESIGN OF HIGH SPEED CARRY SELECT ADDER USING KOGGESTONE ADDER**

Volume 06, Issue 12, Pages: 345–351.
Paper Authors

M.NAVEEN KUMAR, S.DADAPEER
CVRT, AP, India

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

DESIGN OF HIGH SPEED CARRY SELECT ADDER USING KOGGESTONE ADDER

[1]M.NAVEEN KUMAR, [2]S.DADAPEER
[1]PG Scholar, ECE, CVRT, AP, India
[2]Asst Professor, ECE, CVRT, AP, India

Abstract :In this paper, Carry Select Adder (CSA) architecture are proposed using parallel prefix adder. Instead of using 16-bit Ripple Carry Adder (RCA), parallel prefix adder i.e., 16-bit Brent Kung (BK) adder is used to design CSA. Adders are key element in digital design, performing not only addition operation, but also many other function such as subtraction, multiplication and division. Ripple Carry Adder (RCA) gives the most complicated design as-well-as longer computation time. The time critical application use Brent Kung parallel prefix adder to drive fast results but they lead to increase in area. Carry Select Adder understands between RCA and BK in term of area and delay. Delay of RCA is larger therefore we have replaced it with Brent Kung parallel prefix adder which gives fast result. Power and delay of 16-bit RCA and 16-bit BK adder architecture .

INTRODUCTION

Digital filters are very important part of DSP. In fact their extraordinary performance is one of the key reasons that DSP has become so popular. Filters have two uses: signal separation and signal restoration. Signal separation is needed when the signal has been contaminated with interference, noise or other signals. Which is better? Analog filters are cheap, fast and have a large dynamic range both in amplitude and frequency. Digital filters in comparison are vastly superior in the level of performance that can be achieved. Digital filters can achieve thousand of times better performance than an analog filter. This makes a dramatic difference in how filtering problems are approached. With analog filters, the emphasis is on handling limitations of the electronics such as the accuracy and stability of the resistors and capacitors. In comparison digital filters are so good that the performance of the filter is frequently ignored. The emphasis shifts to the limitations of the signals and the theoretical issues regarding their processing. Multiplying a variable by a set of known constant coefficients is a common operation in many digital signal processing (DSP) algorithms. Compared to other common operations in DSP algorithms, such as addition, subtraction, using delay elements, etc., multiplication is generally the most expensive. There is a trade-off between the amount of logic resources used (i.e. the amount of silicon in the integrated circuit) and how fast the computation can be done. Compared to most of the other operations, multiplication requires more time given the same amount of logic resources and it requires more logic resources under the constraint that each operation must be completed within the same amount of time.

A general multiplier is needed if one performs multiplication between two arbitrary variables. However, when multiplying by a known constant, we can exploit the properties of binary multiplication in order to obtain a less expensive logic circuit that is functionally equivalent to simply asserting the constant on one input of a general multiplier. In many cases, using a cheaper implementation for only multiplication still results in significant savings when considering the entire logic circuit because multiplication is relatively expensive. Furthermore, multiplication could be the dominant operation, depending on the application.Multipliers are key components of many high performance systems such as FIR filters, microprocessors, digital signal processors, etc. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest clement in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area-speed constraints have been designed with fully parallel. Multipliers at one end of the spectrum and fully serial multipliers at the other end. In between are digit serial multipliers where single digits consisting of several bits are operated on. These multipliers have moderate performance in both speed and area. However, existing digit serial multipliers have been Plagued by complicated switching systems and/or irregularities in design. Radix 2^n multipliers which operate on digits in a parallel fashion instead of bits bring the pipelining to the digit level and avoid most of'the above problems. They were introduced by M. K. Ibrahim in 1993. These structures are iterative and modular. The pipelining done at the digit level brings the benefit of constant operation speed irrespective of the size of' the multiplier. The clock speed *is* only determined by the digit size which is already fixed before the design is implemented.

Although binary calculations are the dominant in most machines, they are not suitable for commercial, banking, and business applications due to the unacceptable inexact decimal to binary conversion errors they produce. A real example shows the extreme effect of these wrong approximations, where it stated that if a communication company approximates a 5% sales tax on an item (such as a $0.70 telephone call), the yearly loss is over than a $5 million.In Processors (DSP) and microprocessor data path units, adder is an important element. As such, extensive research continues to be focused on improving the power-delay performance of the adder. In VLSI implementations, parallel-prefix adders are known to have the best performance. Reconfigurable logic such as Field Programmable Gate Arrays (FPGAs) has been gaining in popularity in recent years because it offers improved performance in terms of speed and power over DSP-based and microprocessor-based solutions for many practical designs

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

involving mobile DSP and telecommunications applications and a significant reduction in development time and cost over Application Specific Integrated Circuit (ASIC) designs. The power advantage is especially important with the growing popularity of mobile and portable electronics, which make extensive use of DSP functions. However, because of the structure of the configurable logic and routing resources in FPGAs, parallel-prefix adders will have a different performance than VLSI implementations. In particular, most modern FPGAs employ a fast-carry chain which optimizes the carry path for the simple Ripple Carry Adder (RCA).

PARALLEL PREFIX ADDERS

Binary addition is the most fundamental and frequently used arithmetic operation. A lot of work on adder design has been done so far and many architectures have been proposed. When high operation speed is required, tree structures like parallel-prefix adders are used. The Parallel Prefix addition is done in three steps, which is shown in fig1. The fundamental generate and propagate signals are used to generate the carry input for each adder. Two different operators black and gray are used here.

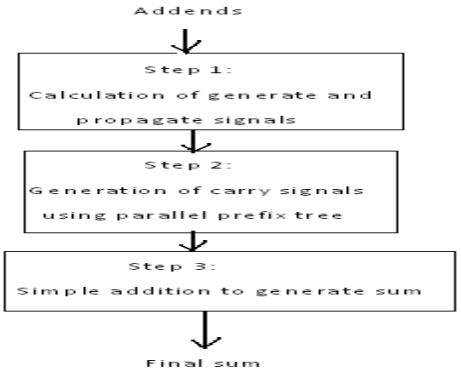


Fig 1. Addition procedure using Parallel Prefix tree structures

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:

$$Sum_i = a_i \wedge b_i \wedge carry_i \quad (1)$$

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:

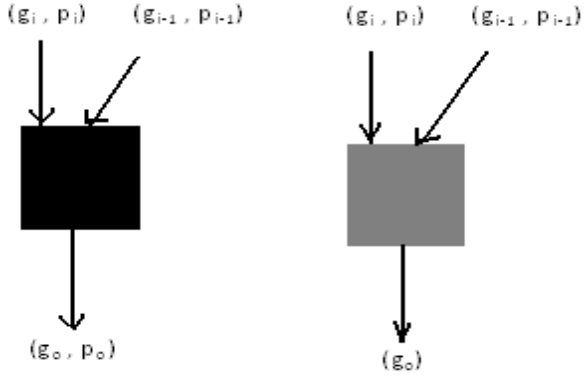
$$G_i = a_i \wedge b_i \quad (2)$$

$$P_i = a_i \wedge b_i \quad (3)$$

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

$$C_{i+1} = g_i \vee (p_i) \quad (4)$$

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.



(a) black operator (b) gray operator

Fig 2 Operators used in Parallel Prefix trees

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

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

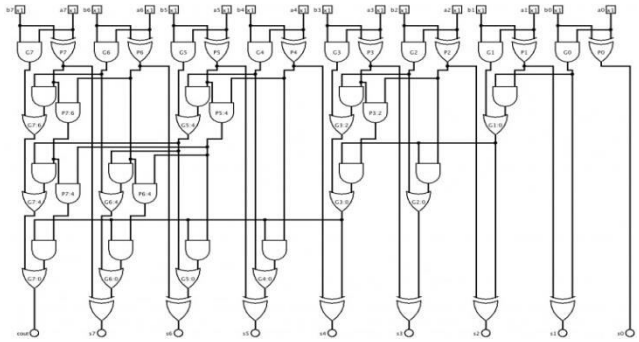


Fig 3 16 bit Kogge-Stone adder

BRENT KUNG

The Brent Kung adder computes the prefixes for 2 bit groups. These prefixes are used to find the prefixes for the 4 bit groups, which in turn are used to compute the prefixes for 8 bit groups and so on. These prefixes are then used to compute the carry out of the particular bit stage. These carries will be used along with the Group Propagate of the next stage to compute the Sum bit of that stage. Brent Kung Tree will be using *2log2N - 1* stages. Since we are designing a 32-bit adder the number of stages will be 9. The fanout for each bit stage is limited to 2. The diagram below shows the fanout being minimized and the loading on the further stages being reduced. But while actually implemented the buffers are generally omitted.

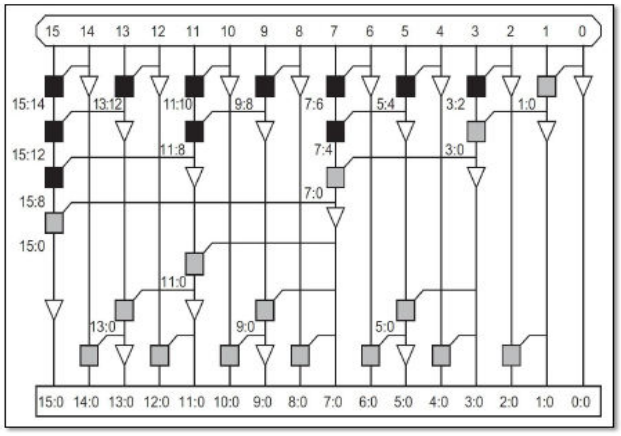


Figure 16-bit Brent Kung Adder

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

CARRY SELECT ADDER

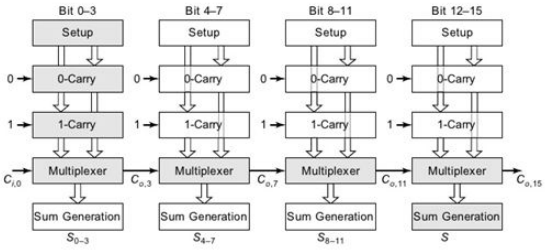
The carry select adder is constructed by cascading each single-bit full-adder. In the carry ripple adder, each full-adder starts its computation till previous carry-out signal is ready. Therefore, the critical path delay in a carry ripple adder is determined by its carry-out propagation path. For an N-bit full-adder as illustrated in Fig. 6.1, the critical path is N-bit carry propagation path in the full-adders. As the bit number N increases, the delay time of carry ripple adder will increase accordingly in a linear way. In order to improve the shortcoming of carry ripple adder to remove the linear dependency between computation delay time and input word length, carry select adder is presented. The carry select adder divides the carry ripple adder into M parts, while each part consists of a duplicated (N/M)-bit carry ripple adder pair, as illustrated in Fig. 6.2 as M=16 and N=4. This duplicated carry ripple adder pair is to anticipate both possible carry input values, where one carry ripple adder is calculated as carry input value is logic "0" and another carry ripple adder is calculated as carry input value is logic "1". When the actual carry input is ready, either the result of carry "0" path or the result of carry "1" path is selected by the multiplexer according to its carry input value. An example of 5-bit carry select adder is illustrated in Fig. 6.3. To anticipate both possible carry input values in advance, the start of each M part carry ripple adder pair no longer need to wait for the coming of previous carry input. As a result, each M part carry ripple adder pair in the carry select adder can compute in parallel. In this way, the critical path of N bit adder can be greatly reduced. In the conventional N-bit carry ripple adder design, the critical path is N-bit carry propagation path plus one summation generation stage. Alternatively, the critical path is (N/M)-bit carry propagation path plus M stage multiplexer with one summation generation stage in the N-bit carry select adder. Since M is much smaller than N and delay in the multiplexer is smaller than that in the full adder, the computation delay in the carry select adder is much shorter than that in the carry ripple adder. However, implementing the adder with duplicated carry generation circuit costs almost twice hardware and twice power consumption as compared with the carry ripple adder.

16-B REGULAR CSLA

The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripple carry adders). In order to perform the calculation twice, one time with the assumption of the carry-in being zero and the other assuming it will be one. After the two results are calculated, the correct sum, as well as the correct carry-out, is then selected with the multiplexer once the correct carry-in is known. The number of bits in each carry select block can be uniform, or variable. In the uniform case, the optimal delay occurs for a block size of $\sqrt{n}$. When variable, the block size should have a delay, from addition inputs A and B to the carry out, equal to that of the multiplexer chain leading into it, so that the carry out is calculated just in time. The $\sqrt{n}$ delay is derived from uniform sizing, where the ideal

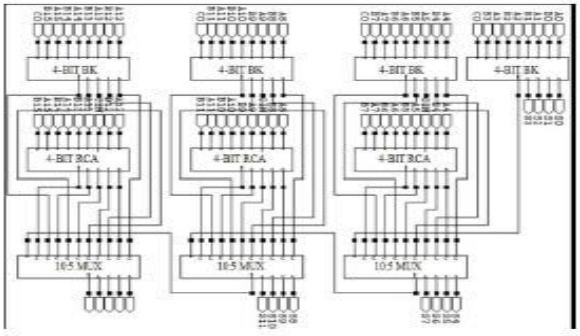
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

number of full-adder elements per block is equal to the square root of the number of bits being added, since that will yield an equal number of MUX delays.



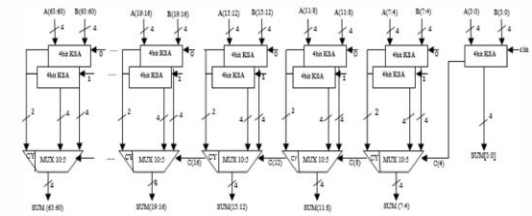
BRENT KUNG CARRY SELECT ADDER

Conventional Carry Select Adder consists of dual Ripple Carry Adders and a multiplexer. Brent Kung Adder has reduced delay as compared to Ripple Carry Adder. So, Regular Linear BK CSA is designed using Brent Kung Adder. Regular Linear KS CSA consists of a single Brent Kung adder fo r Cin=O and a Ripple Carry Adder for Cin=1. It has four groups of same size. Each group consists of single Brent Kung adder, single RCA and multiplexer. We use tree structure form in Brent Kung adder to increase the speed of arithmetic operation. The schematic diagram of Regular Linear BK CSA is shown in Fig. 3.



KOGGE STONE CARRY SELECT ADDER

The CSLA is used in many computational systems design to moderate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. It uses independent ripple carry adders (for Cin=O and Cin=l) to generate the resultant sum. However, the Regular CSLA (RCSLA) is not area and speed efficient because it uses multiple pairs of Kogge Stone Adders (KSA) to generate partial sum and carry by considering carry input. The final sum and carry are selected by the multiplexers (mux). Due to the use of two independent KSA the area will increase which leads an increase in delay. To overcome the above problem, the basic idea of the proposed work is to use n-bit first zero finding logic. This logic can be replaced in KSA for "Cin=l" to further reduces the area. Using a fust zero finding logic instead of KSA in the RCSLA will achieve lower area of Modified CSLA (MCSLA). The main advantage of this zero finding logic comes from the lesser number of logic gates than the Full Adder (FA) structure because the number of gates used will be decreased.



The structure of 64 bit Linear Kogge Stone Carry Select Adder is shown in Fig.2. It has sixteen groups of same size KSA. Each group consists of two identical 4 bit Kogge stone adders and one 10:5 multiplexer except fust group which has single 4bit KSA only. In which we have given Cin=O to one 4bit KSA and Cin=1 to another 4bit KSA. Depending upon the previous carry the selection of either one of the 4bit KSA output is fed to the 10:5 multiplexer along

with carry.Methodology for delay and area evaluations are same for Kogge Stone Linear Carry Select Adder with Cin=O and Cin=1 .Depending upon the selection input i.e carry from previous group, final sum and carry differ in delays[5]. Where as Area evaluation for each group except group 1 remains same.

CONCLUSION AND FUTURE SCOPE

This work can be extended for higher number of bits also. By using parallel prefix adder, delay and power consumption of different adder architectures is reduced. As, parallel prefix adders derive fast results therefore Brent Kung adder is used. The calculated results conclude that BK Carry Select Adder is better in terms of power consumption and high speed when compared with RCA adder architectures and can be used in different applications of adders like in multipliers, to execute different algorithms of Digital Signal Processing like Finite Impulse Response, Infinite Impulse Response etc.

REFERENCES

[1] Sudheer Kumar Yezerla, B Rajendra Naik. "Design and Estimation of delay, power and area for Parallel prefix adders" Proceedings of 2014 RAECS UIET Panjab University Chandigarh, 06 - 08 March, 2014.

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

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

[4] Y. Choi, "Parallel Prefix Adder Design", Proc. 17th IEEE Symposium on Computer Arithmetic, pp. 90-98, 27th June 2005.

[5] Basant Kumar Mohanty and Sujit Kumar Patel "Area–Delay–Power Efficient Carry-Select Adder," IEEE transaction on circuits and systems-II: Express briefs, VOL. NO. 6, JUNE 2014.

[6] L. Mugilvannan and S. Ramasamy "Low-Power and Area-Efficient Carry Select Adder Using Modified BEC-1 Converter" IEEE-31661. 4th ICCCNT2013 July 4-6, 2013, Tiruchengode, India.

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

[8] D. Jaya Kumar, Dr.E. Logashanmugam, ―Performance Analysis of FIR filter using Booth Multiplier‖, IEEE July 2014.

[9] Shelja Jose, Shereena Mytheen ,‖Modified Booth Multiplier Based Low-Cost FIR Filter Design" International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 3, Issue 5, September 2014.

[10] Sarita Chouhan1,Yogesh Kumar2, ―Low power designing of FIR filters‖, ISSN No: 2250-3536 ,May 2012.

[11] Rashidi B, Pourormazd M ―Design and implementation of low power digital FIR filter based on low power multipliers and adders on xilinx FPGA‖,IEEE April 2011.

[12] Ravikumar A Javali, Ramanath J Nayak, Ashish M Mhetar, Manjunath C Lakkannavar‖ Design of High Speed Carry Save Adder using Carry Lookahead Adder "Proceedings of International Conference on Circuits, Communication, Control and Computing (I4C 2014).

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org