RESEARCH PAPER | Engineering | Volume : 5 | Issue : 10 | October 2015 | ISSN - 2249-555X

Comparitive Analysis of Brent-Kang & Kogge- Stone Parallel-Prefix Adder for Their Area, Delay & Power Consumption

KEYWORDS: Parallel-Prefix Adder, Area, Power, Delay.

M.Prasannakumar	V.Sidharthan	Dr. K. Gopalakrishnan
Asst. Professor Dept of Electronics, SNR SONS College, Coimbatore-6.	Asst. Professor Dept of Electronics, SNR SONS College, Coimbatore-6.	Research Supervisor Dept of Electronics, SNR SONS College, Coimbatore-6.

ABSTRACT *Parallel Prefix adders have been one of the most notable among several designs proposed in the past. Parallel Prefix adders (PPA) are family of adders derived from the commonly known carry look ahead adders. The need for a Parallel Prefix adder is that it is primarily fast when compared with ripple carry adders. The classical parallel prefix adder structures presented in the literature over the years optimize for logic depth, area, and fan-out and interconnect count of logic circuits. In this paper, a comparison of two 8- bit parallel-Prefix adder (BENT-KANG AND KOGGE-STONE) in their area, delay, power is proposed. In this proposed system Bent-kang and Kogge-stone parallel prefix adder are used for comparison. The results reveal that the proposed Bent-kang Parallel-Prefix adder is more competent than Kogge-stone Parallel-Prefix adder in terms of area, delay & power. Simulation results are compared and verified using Xilinx 8.1i software.*

Introduction

The arithmetic operations of binary numbers are one of the most interesting problems in modern digital VLSI systems consuming a major design effort of digital signal processors and general purpose microprocessors. The design of high-speed, low-power and area efficient binary adders always receives a great deal of attention. Among the hundreds adder architectures known in the literature, when high performances are mandatory, parallel prefix trees are generally preferable[1]. Parallel Prefix Adders have been

established as the most efficient circuits for binary addition in digital systems. Their regular structure and fast performance makes them particularly attractive for VLSI implementation. The delay of a parallel prefix adder is directly proportional to the number of levels in the carry propagation stage.

The need for a Parallel Prefix adder is that it is primarily fast when compared with ripple carry adders. Parallel Prefix adders have been established as the most efficient circuits for binary addition. These adders are best suited for adders with wider word lengths.

This paper investigates the performance of two different parallel prefix adders. The parallel prefix adders investigated in this paper are: Brent Kung Adder & Kogge Stone Adder. The performance metrics considered for the analysis of the adders are: area, delay and power. Using simulation studies area, delay and power performance of the various adder modules were obtained. It was observed that Kogge Stone Prefix tree adder has better circuit characteristics in terms of delay compared to adders realized using other algorithms.

The rest of the paper is organized as follows: In Section 2 a brief description of the two different parallel prefix adders are given, in Section 3, the methodology used for the research is explained. Section 4 gives results analysis and Section 5 gives conclusions.

1. Parallel- Prefix Adder

Parallel prefix adders are constructed out of fundamental carry operators denoted by ¢ as follows:

$(G_{\cdot\cdot}, P_{\cdot\cdot}) \text{ ¢ } (G_{\cdot}, P_{\cdot}) = (G_{\cdot\cdot}+G_{\cdot}\cdot P_{\cdot\cdot}, P_{\cdot}\cdot P_{\cdot\cdot})$,

where P'' and P' indicate the propagations, G'' and G' indicate the generations. The fundamental carry operator is represented as Figure 1.

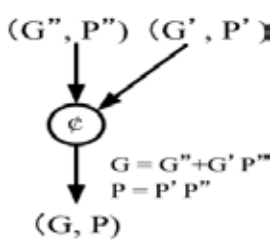
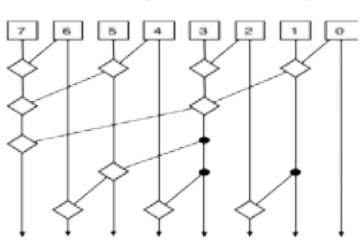


Figure 1. Carry operator

A parallel prefix adder can be represented as a parallel prefix graph consisting of carry operator nodes.

1.1 **Brent Kung Adder**

In 1982, Brent & Kung described this clever modification, which just computes the left-most column in a binary tree, and then fills in the intermediate columns in a reverse tree. The Brent Kung Adder tree diagram is shown in figure 2.



Volume : 5 | Issue : 10 | October 2015 | ISSN - 2249-555X

Figure 2. Brent Kung Adder Tree Diagram

The large number of levels in Brent Kung Adder (BKA) however reduces its operational speed. BKA is also power efficient because of its lowest area delay with large number of input bits [4]. The delay of BKA is equal to (2*log2n)-2 which is also the number of stages for the "o" operator. The BKA has the area (number of "o" operators) of (2*n)-2-log2n where n is the number of input bits [2]. The BKA is known for its high logic depth with minimum area characteristics [5]. High logic depth here means high fan-out characteristics.

2.2 Kogge Stone Adder

In 1973, probably while listening to a Yes or King Crimson album, Kogge and Stone came up with the idea of parallel-prefix computation. The Kogge Stone Adder (KSA) has regular layout which makes them favoured adder in the electronic technology. The Kogge Stone Adder tree diagram is shown in figure 3.

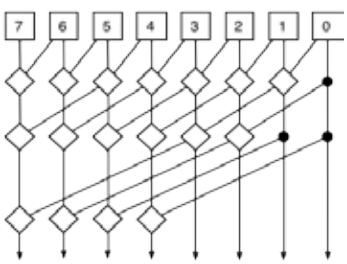


Figure 3. Kogge Stone Adder Tree Diagram

Another reason the KSA is the favoured adder is because of its minimum fan-out or minimum logic depth. As a result of that, the KSA becomes a fast adder but has a large area [6]. The delay of KSA is equal to log2n which is the number of stages for the "o" operator. The KSA has the area (number of "o"operators) of (n*log2n)-n+1 where n is the number of input bits [3].

2. Methodology :

The designs for the adders were produced by writing Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) source file. Fig. 4 shows the system design flow chart. The VHDL source code writing is the most important part in this project. There are a total of two VHDL source codes for 8 bits Brent Kung and Kogge Stone adders. For this research, the VHDL source codes contain elements such as entity, library, architecture, function and array. The design file has to be analyzed, synthesis and compile before it can be simulated. Simulation results in this project come in the form of Register Transfer Level (RTL) diagram and functional waveform. The synthesis report obtained in the form of gate count for the design (area), delay report and power consumption report. The RTL design can be obtained by using the RTL viewer based on the Net list viewer.

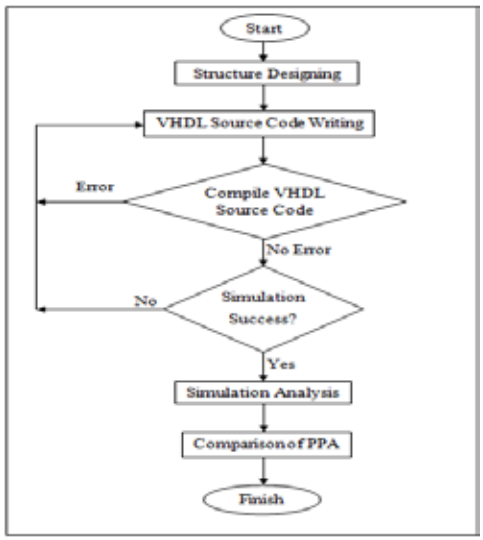


Figure. 4 system design flow chart

Finally, the PPA comparison will be made once the two simulation results are analyzed. BKA and KSA will be compared at this stage. The comparisons will be based on the power consumption, propagation delay and area.

3. Results Analysis :

The BKA and KSA are compared in three main aspects, area, delay and power. The comparison for area, delay and power consumed by the adder circuits are based on tree diagram of each adder. The results are shown in tabulation.

Table. 1 Comparison of BKA and KSA

S.no	Specifications	Brent Kung Adder	Kogge Stone Adder
1.	Gate count(area)	671	924
2.	Delay	3.229ns	3.409ns
3.	Power consumption	39mW	40mW

4. Conclusion:

In terms of area between the two PPAs, the BKA proves to be a better choice. Even though the BKA's area rises as the bit size increase, it does not rise as drastically as KSA. The higher the number of bits supported by the PPAs, the bigger is the adder in terms of area. In terms of computational delay also BSK is better in time propagation delay (tpd) for the bit size of 8- bits, In terms of total power estimated also the BSK consumes less amount of power. Therefore the results reveal that the proposed Bent-kang Parallel-Prefix adder is more competent than Kogge-stone Parallel-Prefix adder in terms of area, delay & power for the bit size of 8bits.

REFERENCE [1] Sivannarayana Gandikota, Raveendra babu Maddasani and Padmasree CH. "Design and Implementation of Carry Tree Adders using Low Power FPGAs", International Journal of Advanced Research in Computer Engineering & Technology , September 2012, pp no. 295- 299. [2] R. P Brent & H. T. Kung, "A Regular Layout for Parallel Adders," IEEE Trans. Computers, Vol C-31, pp 260-264, 1982. [3] P. M. Kogge & H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," IEEE Trans.Computers, Vol. C-22, pp 786-793, 1973. [4] V. Ionescu, I. Bostan, & L. Ionescu, "Systematic Design for Integrated Digital Circuit Structures" IEEE Journal of Semiconductor Conference,2004, Volume 2, pp 467 – 470, 2004. [5] M. M. Ziegler & M. R. Stan, "A Unified Design Space for Regular Parallel Prefix Adders" IEEE Journal of Design, Automation and Test in Europe Conference and Exhibition, Volume 2, pp 1386 - 1387, 2004. [6] I. Koren, Computer Arithmetic Algorithms Ak Peters Series, Second Edition, Massachusetts: A K Pete.