International Journal for Modern Trends in Science and Technology
Volume 10, Issue 04, pages 37-40.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol10issue04.html
DOI: https://doi.org/10.46501/IJMTST1004006

Design and Implementation of High Speed Adders using Fast Fourier Transform

S Aruna kumari[1], S Gunavardhini[2], Shusitha Mucherla[2], V kamalnadh[2], T Vinod Sahu[2]

[1]Assistant Professor, Raghu Engineering college (A), Visakhapatnam, India.
[2]Raghu Engineering college (A), Visakhapatnam. India.

To Cite this Article
S Aruna kumari, S Gunavardhini, Shusitha Mucherla, V kamalnadh, T Vinod Sahu, Design and Implementation of High Speed Adders using Fast Fourier Transform, International Journal for Modern Trends in Science and Technology, 2024, 10(04), pages. 37-40.https://doi.org/10.46501/IJMTST1004006

Article Info
Received: 18 March 2024; Accepted: 03 April 2024; Published: 04 April 2024.

ABSTRACT

Fourier Transform (FFT) is an important signal processing technique widely used in various applications, such as audio and image processing, telecommunications, and scientific computing [1]. Many electronic applications require the best FFT with the least area, low power consumption, high speed. To design an FFT, a multiplier and adder are required. To perform multiplication, adder is the important block. So, if addition is done faster, FFT also work faster. The working of various adders like ripple-carry adder, Sklansky adder, Kogge-Stone adder, and Brent-Kung adder is described in this thesis. By using these various adders working of FFT is also described in this thesis. The ripple-carry adder, Sklansky adder, Kogge-Stone adder, and Brent-Kung adder are implemented in Verilog using Cadence tools. By using these various adders, FFT using RCA, FFT using Sklansky adder, FFT using Kogge-Stone adder, and FFT using Brent-Kung adders are implemented in Verilog using Cadence tools. The performance of various FFT's is compared in terms of cell count, power, and delay.

KEYWORDS: FFT, RCA, Sklansky adder, Kogge-stone adder.

1. INTRODUCTION

The Fast Fourier Transform (FFT) is a mathematical algorithm used to compute the Discrete Fourier Transform (DFT) of a sequence of data in a faster and more efficient way than traditional methods. The DFT is a mathematical technique that transforms a signal from the time domain to the frequency domain. The FFT uses a divide-and-conquer approach to break down the DFT into smaller problems, which are solved recursively and combined to produce the final DFT output. It has found extensive applications in signal processing, image processing, audio processing, and communications. By using the FFT, the signals and data to extract valuable information such as frequency content and spectral characteristics can be analyzed. The FFT has become an essential tool in modern technology and is crucial for professionals working in fields related to signal processing and analysis.

A parallel prefix adder, also known as a carry-look ahead adder, is a digital circuit used to add two or more binary numbers in parallel. The principle of a parallel prefix adder is based on the concept of carry

propagation. The carry propagate function (P) and the carry generate function (G) are two important functions used in the design of parallel prefix adders. These functions are used to generate the carry signals that are required to add two binary numbers in parallel.

Literature Survey:

1. S. K.C, S. M., G. B.C., L. D.M., Navya and P. N.V, in their paper "Performance Analysis of Parallel Prefix Adder for Datapath VLSI Design," described about four types of Parallel prefix adders (PPA), Sklansky adder, Kogge-Stone adder, Brent-Kung adder and Ladner-Fischer adder, and their performance analysis of PPA considered on area, delay and power consumption and simulation for 8-bit input data width. [1]
2. S. Daphni and K. S. V. Grace, in their article described about the design and analysis of various parallel prefix adders (PPA) and compared the performances of these adders on the aspects of area, delay, and power. They also described that the Kogge stone adder (KSA) is preferable for the delay process, increasing the speed of addition automatically. Yet, it requires more space and power. [2]
3. U Penchalaiah and Siva Kumar VG, in their article, described about the design and developed a new PPA architecture, namely the Kogge Stone Adder (KSA) for 8, 16, 32, and 64-bit addition. They also described on the implementation and the results were compared with CSKA in terms of area, delay, speed, and power consumption. [3]
4. Sudheer Kumar Yezerla and B. Rajendra Naik, in their article, implemented all adders in Verilog Hardware Description Language (HDL) using Xilinx Integrated Software Environment (ISE) 13.2 Design Suite, and described on the delay measures using a logic analyzer for all these adders' delay, power, and area. [4]
5. Aradhanan Raju and Sudhir Kumar Sa, in their article, choose to implement the fastest PPA, i.e., KSA, to get a comparative idea about the performance of four different multipliers, namely, the binary multiplier, the Braun multiplier, the Vedic multiplier, and the Baugh Wooley multiplier, based on the better adder of their area, delay, and power. [5]

Existing Method:

The parallel prefix adder operates by dividing the binary numbers into groups of bits, and then processing each group in parallel. The carry signals generated by the P and G functions are used to compute the final sum. In pre-processing stage, generate and propagate logic are included. This generate and propagate logic can be defined as Propagate/generate generator. The propagate generator XORs the input bits together to determine if this bit position propagates a previous carry. The result of the propagate generator is used directly in generating the sum bits. The generate generator ANDs input bits together in order to determine if this bit position will generate a new carry. The standard XOR and gates used are the same as in the boolean unit. In carry generation stage two logics namely black cell logic and grey cell logic are responsible to obtain the desired output. Both black cell and grey cell have same AND OR gate configuration

Proposed Design:

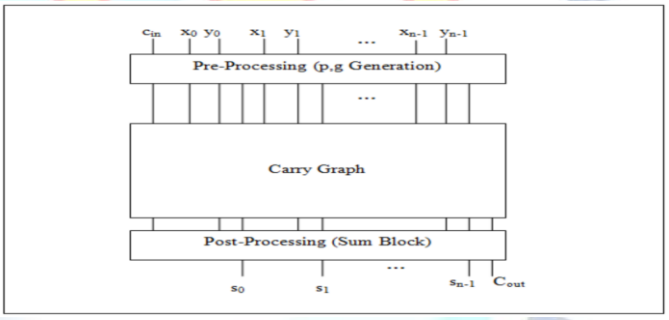

Fig: Parallel Prefix Adder Mechanism

Carry generation stage works by creating two signals (p and g) for each bit position, based on if a carry is propagated through from a less significant bit position (at least one input is a '1'), a carry is generated in that bit position (both inputs are '1'), or if a carry is killed in that bit position (both inputs are '0'). In most cases, p is simply the sum output of a half-adder and g is the carry output of the same adder. After p and g are generated the carries for every bit position are created. This permits the adders to work simultaneously.

The Sklansky adder operates by subdividing the binary addition problem into smaller problems that can each be resolved on its own. Instead of doing the addition in order, the adder does it in parallel using a structure like a tree. The Sklansky adder computes a new carry bit at each level of the tree using the input bits and

carry bits from the level before. The following level of carries are then computed using this carry bit, and so on up the tree until the last carry bit is computed. The carriers and the sum bits are computed simultaneously. The Sklansky adder has a time complexity of O(log n), providing a speedup compared to a ripple carry adder, which has a time complexity of O(n), when adding multiple bits in parallel. Where n is the total number of bits being added.

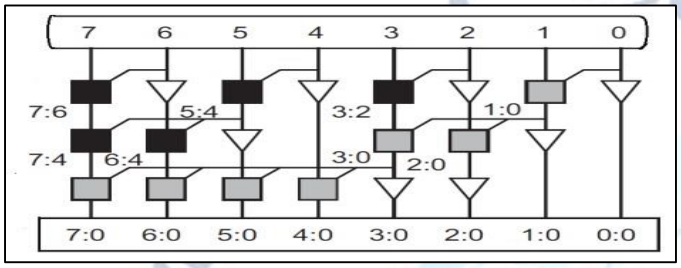

Fig: Algorithm for 8-bit Sklansky Adder

Another kind of parallel prefix adder is the Kogge-Stone adder, also called a ripple-carry tree adder. The Kogge-Stone adder uses a tree-like structure, just like the Sklansky adder does, to carry out binary addition in parallel. The Kogge-Stone adder, as opposed to the Sklansky adder, propagates the carry bits across the tree using a ripple-carry method. The Kogge-Stone adder operates by grouping the input bits and parallelizing the carry bit computations for each group. The carry bits for each group are then used to calculate the carry bits for the following group, and so on up the tree until the last carry bit is calculated. The sum bits are also computed in parallel. Similar to the Sklansky adder, the Kogge-Stone adder has an O (log n) time complexity.

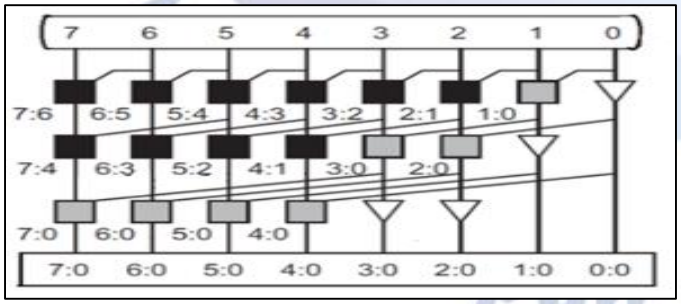

Fig: Algorithm for 8-bit Kogge-stone Adder

The Brent-Kung adder is a different kind of parallel prefix adder. It is similar to the Sklansky and Kogge-Stone adders which uses a tree-like structure to conduct binary addition in parallel. The carry bits for each group are simultaneously computed by the Brent-Kung adder after grouping the input bits. The Brent-Kung adder, on the other hand, combines both a ripple-carry mechanism and a carry-look ahead mechanism. The Brent-Kung adder uses a ripple-carry mechanism to calculate a partial sum and a partial carry at each level of the tree.

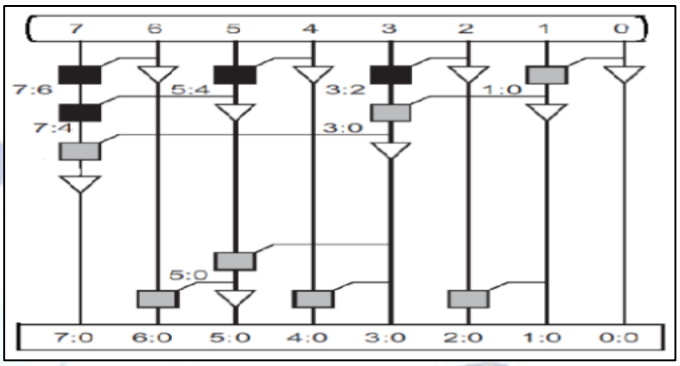

Fig: Algorithm for 8-bit Brent-Kung Adder

Simulation Results:

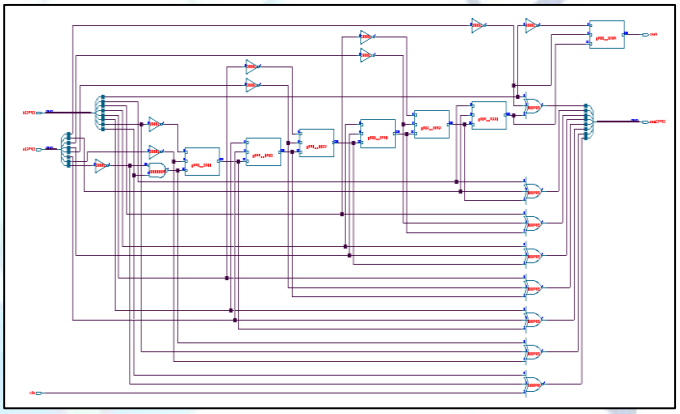

Fig: RTL Schematic of Sklansky Adder

Sl. No	FFT using various Adders	Cells	Total Power (in mW)	Delay (in ps)
1	Ripple Carry Adder	458	241.870	1268
2	Kogge-Stone Adder	474	247.826	878
3	Sklansky Adder	474	246.918	878
4	Brent-Kung Adder	484	252.279	721

Fig: Comparison Table for various adders using FFT C

CONCLUSION:

Implemented various adders like Ripple Carry adder, Sklansky adder, Brent Kung adder and Kogge-Stone adder using Verilog. By using these various adders, FFT was implemented using Ripple carry adder, Sklansky adder, Brent-Kung adder and Kogge-Stone adder in Verilog using Cadence tools. Performance of this FFT using various adders is analyzed in terms of power, delay and cell count. The delay offered by FFT using RCA is more than the other adders since the RCA uses

individual full adders for every bit computation. The FFT using Kogge-Stone adder and FFT using Sklansky adder have better performance than ripple carry adder. FFT with Brent-Kung adder is having a minimal delay than Ripple carry adder, Sklansky adder and Kogge-Stone adder. Also, Power Delay Product of FFT using Brent-Kung adder is least among the designed implementations. So, FFT using Brent-Kung Adder is the most effective implementation which can be used for various designs in signal processing and communications.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] S. K.C., S. M., G. B.C., L. D.M., Navya and P. N.V., "*Performance Analysis of Parallel Prefix Adder for Datapath VLSI Design,*" 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 1552-1555.

[2] S. Daphni and K. S. V. Grace, "*A review analysis of parallel prefix adders for better performance in VLSI applications,*" 2017 IEEE International Conference on Circuits and Systems (ICCS), Thiruvananthapuram, India, 2017, pp. 103-106.

[3] U. Penchalaiah and S. K. VG, "*Design of High-Speed and Energy-Efficient Parallel Prefix Kogge Stone Adder,*" 2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA), Pondicherry, India, 2018, pp. 1-7.

[4] S. K. Yezerla and B. Rajendra Naik, "*Design and estimation of delay, power and area for Parallel prefix adders,*" 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India, 2014, pp. 1-6.

[5] A. Raju and S. K. Sa, "*Design and performance analysis of multipliers using Kogge Stone Adder,*" 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (ICATCCT), Tumkur, India, 2017, pp. 94-99.

[6] A. Prasath A.M., R. V. Arjun, K. Deepaknath and K. Gayathree, "*Implementation of optimized digital filter using sklansky adder and kogge stone adder,*" 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 661-664.

[7] A. N. Borodzhieva, "*Implementation of Kogge-Stone Adders Using Computer-Based Tools,*" 2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2022, pp. 1-6.

[8] N. U. Kumar, K. B. Sindhuri, K. D. Teja and D. S. Satish, "*Implementation and comparison of VLSI architectures of 16-bit carry select adder using Brent Kung adder,*" 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 2017, pp. 1-7.

[9] M. A. A. Amin, M. Kartiwi, M. Yaacob, E. A. Z. Hamidi, T. S. Gunawan and N. Ismail, "*Design of Brent Kung Prefix Form Carry Look Ahead Adder,*" 2022 8th International Conference on Wireless and Telematics (ICWT), Yogyakarta, Indonesia, 2022, pp. 1-6.

[10] Ma Cuimei, H. Chen and Ma Long, "*An efficient design of high-accuracy and low-cost FFT,*" IET International Radar Conference 2013, Xi'an, 2013, pp. 1-4.

[11] Xing Sun, Dongli Qiu, He Chen and Dong Chen, "*An implementation of FFT processor,*" IET International Radar Conference 2013, Xi'an, 2013, pp. 1-4.

[12] B. R. Manuel, E. Konguvel and M. Kannan, "*An area efficient high speed optimized FFT algorithm,*" 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, India, 2017, pp. 1-5.