

# VLSI Architecture for Combined $R_2B$ , $R_4B$ & $R_8B$ FFT using Bidirectional Gate

**Rahul Kumar Singh**

*M. Tech. Scholar*

*Dept. of Electronics and Communication  
TRUBA, Bhopal, India*

**Nishi Pandey**

*Assistant Professor*

*Dept. of Electronics and Communication  
TRUBA, Bhopal, India*

**Prof. Abhishek agwekar**

*Head of Dept.*

*Dept. of Electronics and Communication  
TRUBA, Bhopal, India*

**Abstract:-** Fast Fourier Transform (FFT) is a major part in communication modules and DSP processor. FFT includes large number of arithmetic operation during their operation. For each operation it performs large number of switching operations and consumes large amount of delay. Combined  $R_2B$ ,  $R_4B$  &  $R_8B$  and bidirectional gate encourages reducing the number of switching activity by reducing the number of operations in FFT. An enhanced combined  $R_2B$ ,  $R_4B$  &  $R_8B$  and bidirectional gate is utilized to reduce the number of arithmetic operations in the FFT architecture. The performance of the improved FFT architecture is estimated to find its suitability for the low delay Wireless communication system. An arithmetic operation includes addition, subtraction and multiplication. Each arithmetic operation is designed based on the number of ones and zeros in the input data pattern. Based on the set of input data number of switching operations can be reducing by introducing the enhanced pruning algorithm to the FFT. All design is simulated Xilinx software and calculated simulation parameter i.e. slice, look up table (LUT) and delay.

**Keywords:** -  $R_2B$ ,  $R_4B$ ,  $R_8B$ , SDF, FFT

## I. INTRODUCTION

Developments in the field of wireless communication technology are introducing more challenges as well as opportunities for the researchers in the field of Science and Technology. The tremendous growth of internet traffic, which demands high data transmission rates, motivates the researchers for design of robust wireless networks. Large numbers of researchers are concentrating to re-design wireless communication architecture in algorithmic and structural level considering improving the quality of communication link. The technology developments in communication system and DSP processors have been providing the coexistence of different standards. The services available in the recent communication standards need higher data rates, capacities with co-existence of different standards which have driven the advancement of new generation

multi-standard wireless communication systems with higher demand [1].

The design of a high speed power reduced speed and low size system typically needs extensive computations. The question arises whether the traditional method of multicarrier modulation technique can encourage the system to perform the future wideband systems more effectively. A high data rate of signals of symbol with a wide band signal is distributed to lower level of several signals in multicarrier modulation system. The application of FFT/IFFT in different types of modulation schemes with modern typed of modulations is very important. It is established that, Fast Fourier Transform (FFT) and its Inverse Fast Fourier Transform (IFFT), which are widely used in communication systems, are essential in the field of Digital Signal Processing (DSP) applications. The FFT algorithm and its iterations eliminate the redundant calculations needed in Discrete Fast Fourier Transform (DFT) [2, 3].

There are various algorithms to implement FFT, such as radix-2, radix-4 and radix-8; and by adjusting the length of FFT from 128 to 2048, its use can be extended for different applications. Even further higher radix algorithm can decrease the computational complexity in the designs. Higher the radix, lower the computational complexity. In versatile digital systems, apart from the applications, there are concerns about speed of operation, lower power consumption, minimal truncation error and reduced chip size. FFT and IFFT operation consumes more number of operations like addition, subtraction and multiplication in each stage of operation. Designers always take the challenge to balance the power, speed and area of the system designed in digital system design. Multipurpose chips can perform multiple functions - but in relatively slow speed, whereas special purpose chips can perform new function with higher speeds [4, 5].

Field programmable gate array consists of three types of parts. Logic blocks, I/O blocks, interconnection wires and switches. Logic blocks and I/O blocks are connected to the pins of the package. The logic blocks are arranged in the two dimensional array. The interconnection wires are connected as horizontal and vertical routing path between rows and columns of logic blocks. Wires and programmable channels are arranged in the routing channels. For Xilinx FPGA, Configurable logic block is

the basic block. Configurable logic block consists of number of RAM based look up tables to implement proposed logic, using selection circuitry like mux and flip flops. LUTs are highly powerful to configure and handle combinational logics [6, 7, 8].

## II. PROPOSED METHODOLOGY

The consolidated R<sub>2</sub>B, R<sub>4</sub>B & R<sub>8</sub>B based SDF FFT has been planned in this proposed work. The consolidated Radix of FFT design has a lesser measure of computational way and furthermore improves the exhibitions of FFT processor. SDF design, the info information successions are going through one single way. The butterfly preparing component plays out the calculation on the information. The expansion and deduction activity is done in butterfly components. The changed convey select viper circuit is utilized for snake activity in this engineering. This snake structure is extremely productive in this design. The structure of joined R<sub>2</sub>B, R<sub>4</sub>B & R<sub>8</sub>B FFT is appeared in Fig. 1. The engineering of 16 point SDF FFT is appeared in Fig. 2.

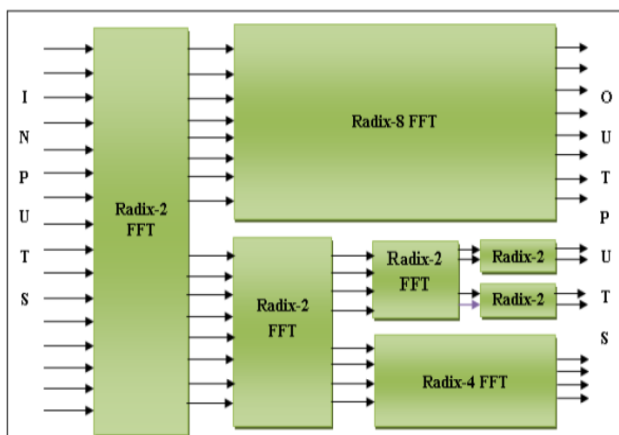


Fig. 1: Structure of Combined  $R_7B$ ,  $R_4B$  &  $R_8B$  FFT

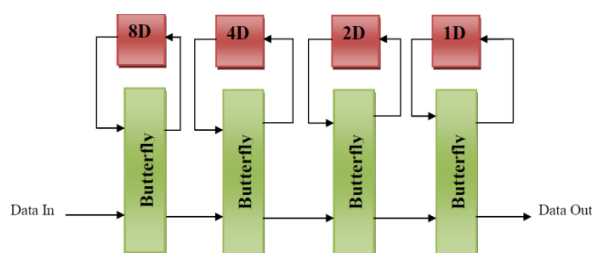


Fig. 2: Architecture of 16 Point SDF FFT

The architecture of improved Radix-8 algorithm involves the resolution of the output terms as complex and real. The inputs are taken as 8-bit vector value for the implementation for the case of real data. The inputs are forwarded using butterfly Radix-2 implementation which helps to improve the combinational delay path. The Radix-2 butterfly module is implemented and instances are called every time for the small computation [2]. For

the simplicity of computation, the variables have been scaled up to overcome the decimal notation wherein the analysis is continued in software wherein the simple division by base 10 will be much faster and efficient in the computation of the final value of in frequency domain [3]. Each stage final values are updated to register and the twiddle factor have been previously computed and stored in register are invoked for the finding the value. This invoked value is multiplied with the value obtained by the Radix-2 instantiation. Each stage value is carried onto the further stage until the final stage is obtained. At the final stage we obtain the complete 8-point Radix-8 FFT for the computation of DFT in optimized and high speed technique. The multiplication is made in the stages of the addition to reduces the area utilization. The complex output is provided as the individual value apart from real values. The final staged values are driven to software or the next macro for further computation and analysis.

## Radix-2

A fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and its inverse. Fourier analysis converts time (or space) to frequency and vice versa; an FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors. As a result, fast Fourier transforms are widely used for many applications in engineering, science, and mathematics.

## Radix-4

In last three decades, various FFT architectures such as single-memory architecture, dual memory architecture, pipelined architecture, array architecture and cache memory architecture have been proposed. In order to improve the power reduction, we propose a radix-4 64-point pipeline FFT/IFFT processor. In order to speed up the FFT computations, more advanced solutions have been proposed using an increase of the radix. The radix-4 FFT algorithm is most popular and has the potential to satisfy the current need.

## Radix-8

Radix-8 is one FFT algorithm which is analyzed and surveyed in order to enhance the throughput of operation by decreasing the calculation and computations; this will be achieved by applying the base to 8. Given same value if base incremented the power must drastically reduce. The value of stages is noticeable reduction by 75% since  $N=82$  i.e., only two stages. By implementation of the FFT algorithm the functional and computational complexity decreases to where indicate the Radix 'r' FFT. Radix 'r' FFT can be progressively obtained from DFT by decomposing the N point DFT into groups of iteratively associated r-point transformation and  $x(n)$  is powers of r. In case of Radix-8 algorithm, r value 8. The DIF Radix-8 FFT repetitively partitions a DFT into eight chunk-lengths DFT of sets of individual eighth sample. The outputs of distributed and smaller FFTs are reutilized into computing

several outputs, which significantly bring down overall computational expense.

### Bidirectional Gate

Several reversible gates have come out in the recent years. The most basic reversible gate is the Feynman gate and is shown in Fig.1. It is the only 2x2 reversible gate available and is commonly used for fan out purposes. Consider the input B as constant. When B is zero, the gate acts as a copying gate or a buffer where both the output lines contain the input A. When B is one, the complement of A is obtained at the output Q. The 3x3 reversible gates include Toffoli gate, Fredkin gate, New gate and Peres gate, all of which can be used to realize various Boolean functions. Fredkin gate is shown in Fig.3. The 4x4 reversible gates include TSG gate, MKG gate, HNG gate, PFAG gate etc.

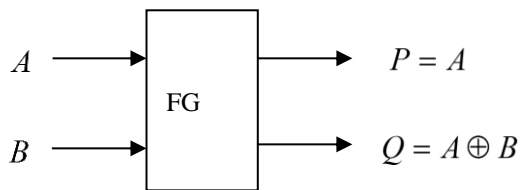


Fig. 3: Feynman gate

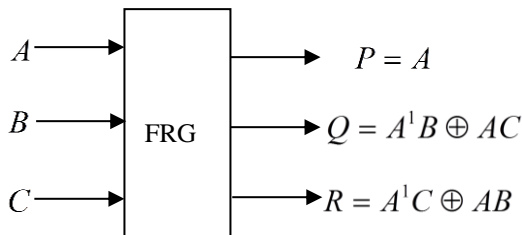


Fig. 4: Fredkin gate

Figure 5 shows the Peres gate. Some of the 3x3 gates are designed for implementing some important combinational functions in addition to the basic functions. Most of the above mentioned gates can be used in the design of reversible adders.

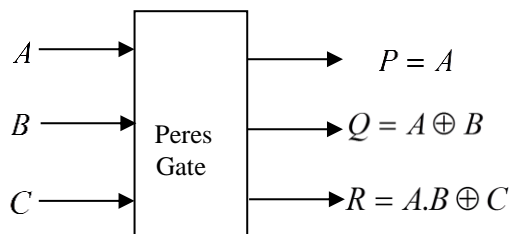


Fig. 5: Peres gate

Several 4x4 gates have been described in the literature targeting low cost and delay which may be implemented in a programmable manner to produce a high number of

logical calculations. The HNG gate, presented in [10], produces the following logical output calculations:

$$P = A \quad (1)$$

$$Q = B \quad (2)$$

$$R = A \oplus B \oplus C \quad (3)$$

$$S = (A \oplus B).C \oplus (AB \oplus D) \quad (4)$$

The quantum cost and delay of the HNG is 6. When  $D = 0$ , the logical calculations produced on the R and S outputs are the required sum and carry-out operations for a full adder. The block diagram representation of the HNG is presented in Fig. 6.

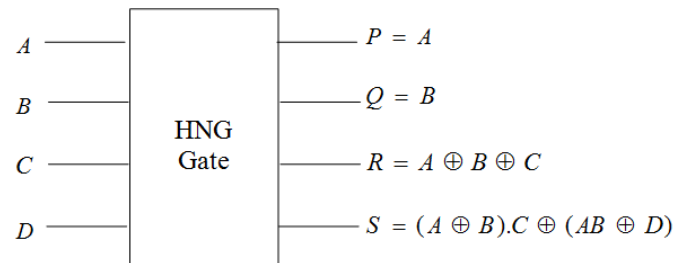


Fig. 6: HNG Gate

The RDKG is presented by figure 7. RDKG is 4x4 system representation. RDKG is working on full adder and full sub-tractor when first input assume '0' and '1'.

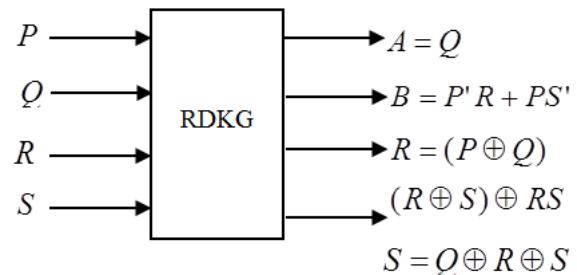


Fig. 7: RDKG Gate

### III. SIMULATION RESULTS

The view technology schematic (VTS) of 32-point FFT using DKG Gate is shown in figure 8.

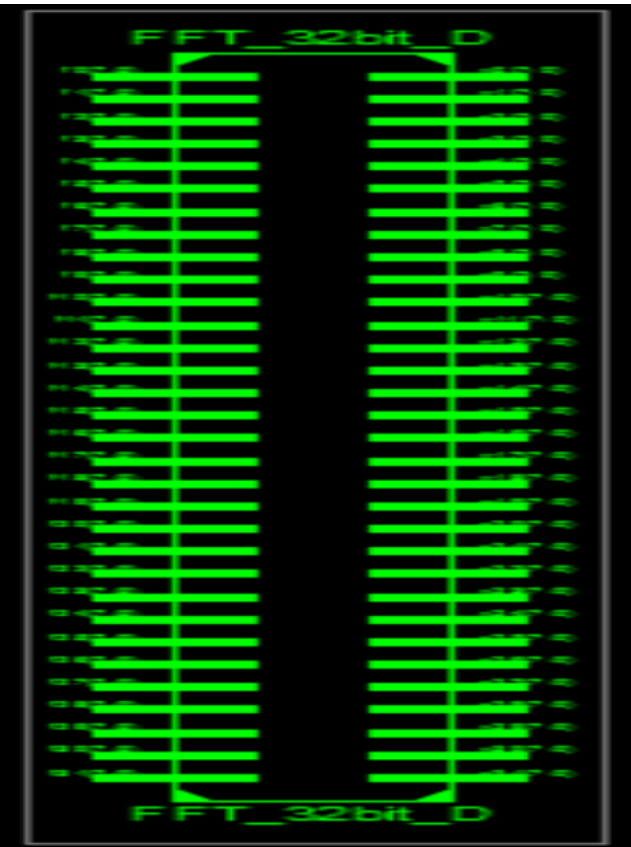


Fig. 8: VTS of 32-point FFT using DKG Gate

This figure 9 is the view technology schematic (VTS) of 32-point FFT. In this, the input is f0-f31 and the out y0-y31.

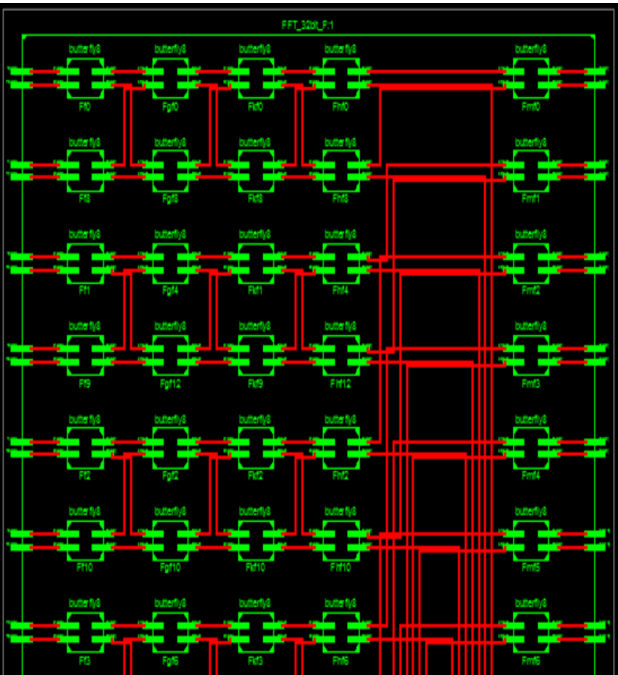


Fig. 9: RTL View of DIT 32-point FFT using DKG Gate

This figure 10 is the RTL Schematic of 32-point FFT it depends on the view technology. It shows all the components inside the VTS.

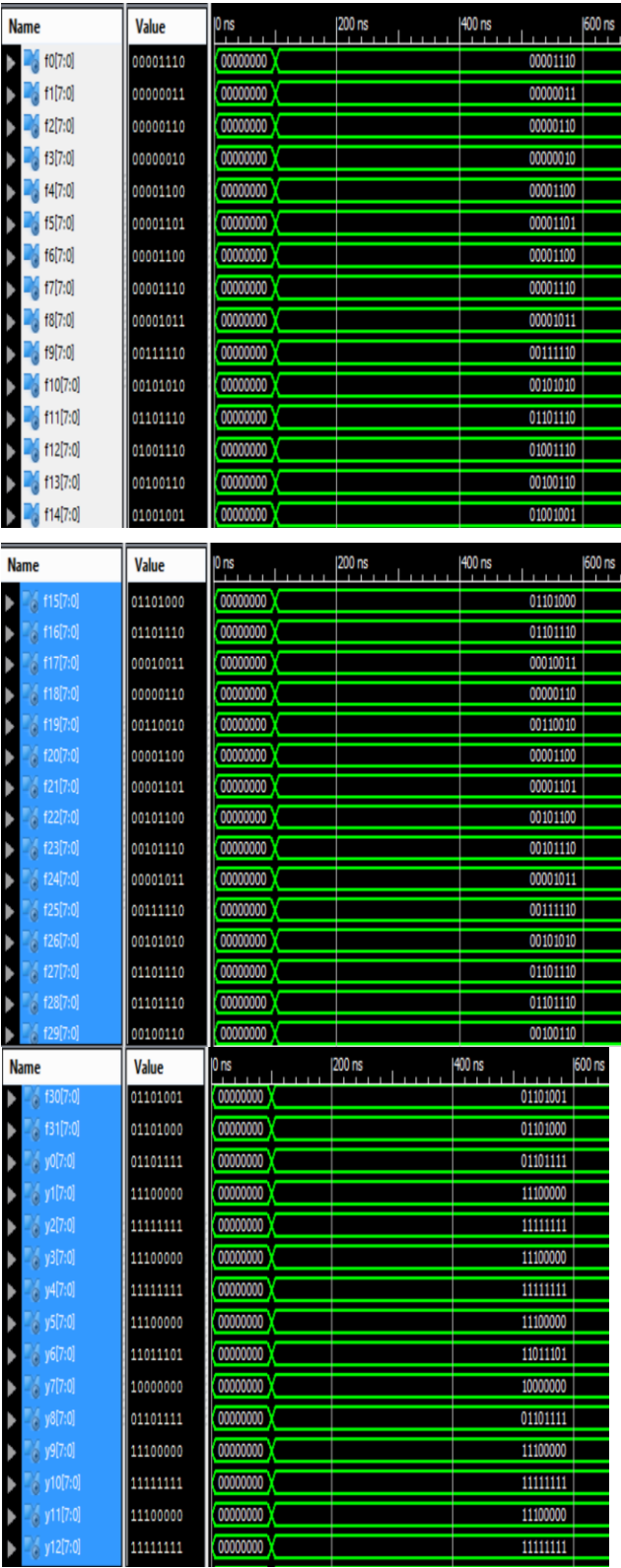


Fig. 10: VTS of DIT 8-point FFT using DKG Gate

This is the VHDL test-bench (VTS) of 32-point FFT. This figure 10 shows the waveform.

Table 1: Comparison Result for Combined SDF FFT with Different Parameter

Design	N	Slices	LUTs	MCPD (ns)
R <sub>2</sub> B SDF FFT	8	145	252	11.378
	16	371	648	12.396
	32	897	1568	13.754
Combined R <sub>2</sub> B, R <sub>4</sub> B and R <sub>8</sub> B based SDF FFT	8	187	336	10.111
	16	514	923	11.425
	32	1158	2069	13.416

#### IV. CONCLUSION

The enhanced combined method is applied to FFT to reduce the delay and area. It is achieved by reducing the number of operations. Number of operations working the proposed method is analyzed, discussed and formulated. A huge reduction in number of operations using the proposed technique. Number of operations is reduced in each combination of input data. Number of operations can be calculated by using each equation formulated during analysis.

#### REFERENCES

- [1] Syeda Farhat Sultana and Basavaraja Patil, "Area efficient VLSI architecture for reversible radix<sub>2</sub> FFT algorithm", International Conference on Emerging Smart Computing and Informatics (ESCI), IEEE 2021.
- [2] Navin Nadar, Moksha Mehta, Kaushal Bhat, Joshua Mendes and Dr. Ravindra Chaudhari, "Hardware Implementation of Pipelined FFT using Polyphase Decomposition", International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), IEEE 2021.
- [3] Shashidhara. K. S and H.C. Srinivasaiah, "Low Power and Area efficient FFT architecture through decomposition technique", International Conference on Computer Communication and Informatics (ICCCI -2017), Jan. 05 – 07, 2019, Coimbatore, INDIA.
- [4] Fahad Qureshi, Jarmo Takala, Anastasia Volkova, Thibault Hilaire, "Multiplier-less Unified Architecture for Mixed Radix-2/3/4 FFTs", 2017 25th European Signal Processing Conference (EUSIPCO), IEEE 2019.
- [5] Guilherme Ferreira; Leandro M. G. Rocha; Eduardo Costa and Sergio Bampi, "Combining m=2 Multipliers and Adder Compressors for Power Efficient Radix-4 Butterfly", 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE 2019.
- [6] S. Aseeri "State-of-the-Art FFT: Algorithms Implementations and Applications" Reflections on the 2018 SIAM Conference on Parallel Processing for Scientific Computing pp. 1-3 Jun 2018.
- [7] E. Konguvel and M. Kannan "A Survey on FFT/IFFT Processors for Next Generation Telecommunication Systems" Journal of Circuits Systems and Computers vol. 27 no. 03 pp. 1830001 2018.
- [8] W. Choi Kang and J. Park "Embedded DRAM- Based Memory Customization for Low-Cost FFT Processor Design" IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2017.
- [9] Fahad Qureshi, Muazam Ali, and Jarmo Takala, "Multiplierless Reconfigurable Processing Element for Mixed Radix-2/3/4/5 FFTs", 978-1-5386-0446-5/17/\$31.00 ©2017 IEEE.
- [10] R. H. Neuenfeld M. B. Fonseca E. A. C. da Costa and J. P. Osés "Exploiting addition schemes for the improvement of optimized radix-2 and radix-4 fft butterflies" 2017 IEEE 8th Latin American Symposium on Circuits Systems (LASCAS) pp. 1-4 2017.
- [11] B. Silveira G. Paim B. Abreu M. Grellert C. M. Diniz E. A. C. da Costa et al. "Power-Efficient Sum of Absolute Differences Hardware Architecture Using Adder Compressors for Integer Motion Estimation Design" IEEE Transactions on Circuits and Systems I: Regular Papers vol. PP no. 99 pp. 1-12 2017.
- [12] M. Najem T. Bollengier J. C. Le Lann and L. Lagadec "A Cost Effective Approach for Efficient Time-Sharing of Reconfigurable Architectures" 2017 International Conference on FPGA Reconfiguration for General-Purpose Computing (FPGA4GPC) May 2017.
- [13] M. Z. A. Khan and S. Qadeer "A new variant of Radix-4 FFT" 2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN) pp. 1-4 2016.
- [14] C. A. Arun and P. Prakasam "A mathematical approach on various radix- 2i FFT algorithms" 2016 International Conference on Electrical Electronics and Optimization Techniques (ICEEOT) pp. 1040-1045 2016.
- [15] A. Rauf M. A. Pasha and S. Masud "A Novel split radix fast fourier transform design for an adaptive and scalable implementation" 2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS) pp. 116-121 2016.