

Design and Implementation of L.U.T by Anti symmetric product coding (APC) and odd-multiple-storage (OMS)

Manasi Vilas Late, Bharathi D.

ECE Dpt. Asst Professor

Jagruti Institute of Engineering and Technology,

JNTU (Hyderabad), Andhra Pradesh, India

manasi.late@rediffmail.com

Abstract: Continual need for high performance in compute bound scientific applications motivates the study of LUT Optimization. LUT Optimization replaces a complex expression with an access to pre computed LUT data containing the expression values. This results in faster expression evaluation and high performance gain. The LUT approach can result in significant improvement in terms of low latency implementation and less dynamic power consumption. Memory based computing structures are more regular than the multiply accumulate (MAC) structures.

Recently, we have proposed the anti symmetric product coding (APC) and odd-multiple-storage (OMS) techniques for lookup-table (LUT) design for memory-based multipliers to be used in digital signal processing applications. Each of these techniques results in the reduction of the LUT size by a factor of two. In this brief, we present a different form of APC and a modified OMS scheme, in order to combine them for efficient memory-based multiplication. The proposed combined approach provides a reduction in LUT size to one-fourth of the conventional LUT.

We have also suggested a simple technique for selective sign reversal to be used in the proposed design. It is shown that the proposed LUT design for small input sizes can be used for efficient implementation of high-precision multiplication by input operand decomposition. It is found that the proposed LUT-based multiplier involves comparable area and time complexity for a word size of 8 bits, but for higher word sizes, it involves significantly less area and less multiplication time.

Keywords – Xilinx ISE, LUT, FIR System, SDR, Spectrum-Sensing, FPGA, Memory- optimization, A-OMS LUT.

INTRODUCTION:

In most of the DSP processors the memory based computing structures are of primary concern than the multiply accumulate structures. Computational or functional operations performed in the DSP blocks of an FPGA for implementing a particular task are time consuming and require more components like adders, multipliers. In the processors like DSP core in FPGAs multiply and accumulate structures are replaced with Look Up Tables. Instead of using conventional multipliers for complex multiplication, operations are simplified with the usage of LUTs that are used for the direct storage of the complex computational values [1, 3]. Further optimization of Look-up-tables provides better performance in terms of speed and effective area utilization. In this paper, LUT optimization using the A-OMS methodology is of primary concern.

Several studies in the past have examined the effect of logic block functionality on the area and performance of field-programmable gate-arrays (FPGAs).

The focus of this paper is to determine the effect of the number of inputs to the LUT. In terms of the algorithms employed, the mappers are divided into structural and functional. Structural mappers consider the circuit graph as given and find a covering of the graph with K-input subgraphs corresponding to LUTs. The functional approaches perform Boolean decomposition of the logic functions of the nodes into sub-functions of limited support size realizable by individual LUTs. Since functional mappers explore a larger solution space, they tend to be time-consuming, which limits their use to small designs [1, 3]. In practice, FPGA mapping for large designs is done using structural mappers, whereas the functional mappers are used for re synthesis after technology mapping.

Secondly, spectrum sensing in the current era of communication technologies is of foremost concern. Most of its applications require an efficient utilization of it. However, according to the recent survey reports, there is still the need to have efficient spectrum sensing techniques. Consider the case of a Software Defined Radio that put forward the concept of cognitive radio to sense the spectrum holes (white spaces) for channel or band reusability. Moreover, in areas like wireless and multimedia applications, digital signal processing applications this efficiency helps to increase the intact system performance. The concept of cognitive radio is considered in SDR to provide a solution for spectrum under utilization where the spectrum holes reuse the channel or band for an authorized spectrum stealing. Here matched filters are considered to be good approach and its structure resembles the FIR filter structure [2, 4]. This can be applicable to different fields of technologies and in wireless and multimedia communication where digital forms of signal processing are now primary concern.

In this paper, A-OMS LUT based FIR filter (that reflects a simple matched filter design as an efficient method for spectrum sensing) is designed for high-speed signal transmissions. A combined approach of the two methods is defined (i.e. Anti symmetric product coding and Odd Multiple Storage that are used previously to optimize LUTs with in a DSP cores for their related operations). The input address and LUT output could always be transformed into odd integers. Previously [5, 6] it is observed that, when an Anti symmetric product coding approach is combined with the Odd multiple storage technique, the two's complement operations could be very much simplified since the input address and LUT output could always be transformed into odd integers, and both cannot be combined since the words generated are odd numbers.

Consequently a different form of Antisymmetric product coding combined with a modified form of Odd Multiple Storage scheme forming A-OMS LUT method which aims mainly to provide the efficient memory based computations and to perform operations for required functional computational. The modified approach is described briefly in the section two of this paper. The

section three consists of an FIR filter based system design with an A-OMS LUT method. In the section, four and five

consists of the memory optimization process, results, and conclusion.

II. A-OMS LUT METHOD

Conventional LUT-based multipliers, with a fixed coefficient and an input word have been used for simple memory based multiplication operations that hoard in a memory core [3]. This requires increase in the LUT size with an increase in the input word length, which is area inefficient. In order to provide an area efficient look-up-table for large data operation, some optimization schemes have been presented, of them in one method, instead of the entire values only the odd multiple values are stored and with another one, there is a reduction in LUT size to half of its original where the product words are recorded as antisymmetric pairs.

Combining the above-specified methods, form A-OMS LUT method that further optimizes the LUTs where modified methods of odd multiple storage and antisymmetric product coding are used.

A. Method 1: Modified anti symmetric product coding scheme: In this method, 32 x 5-bit input words are considered. Computing the product word (PW) values (i.e., input word {X} of length L=5 multiplied by fixed coefficient value A) results in the negative mirror symmetry from half of total input words that facilitates a reduced LUT in size. Hence, for a given 4-bit addresses the corresponding code words to be stored are reduced to half.

This is derived from the anti symmetric behavior of products forming antisymmetric product coding, where the address bits are represented by $x\{x_0, x_1, x_2, x_3\}$ such that $X' = X_L$, if $x_4 = 1$; X_L' , if $x_4 = 0$... (1)

where $X_L = (x_0, x_1, x_2, x_3)$ is the four less significant bits of X, and X_L' is the two's complement of X_L . The product word can be denoted as

$PW = 16A + (\text{sign value}) \times (\text{derived word})$... (2) where sign value is equal to one for $x_4 = 1$ and is equals to -1 for $x_4 = 0$. The product value for $X = (10000)$ corresponds to the derived word i.e., anti symmetric product code value "zero," which could be derived by resetting the LUT output, instead of storing that in the LUT.

A simplified LUT-M circuit for an input word of length L=5 is shown in figure 1.it describes both the structure and function of LUT-M (look-up-table based multiplier).

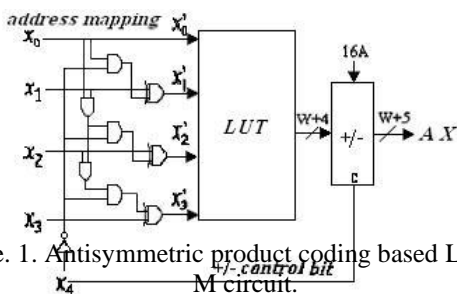


Figure 1. Antisymmetric product coding based LUT-M circuit.

The address mapping circuit generates the desired address $\{x_0', x_1', x_2', x_3'\}$ where x_4 is a control bit for the (+/-)_cell. The address bit generated through address

mapping selects the required value in LUT input whose output then add/subtracted from 16A, by the (+/-)_cell as shown in figure 1.

B. Method 2: Modified odd multiple storage scheme: In this method a barrel shifter is used to perform the shift operations through which the even multiples are computed from the obtained odd multiple values by simple shift operations which provides LUT optimized in terms of area by storing only the odd multiple values rather than whole values.

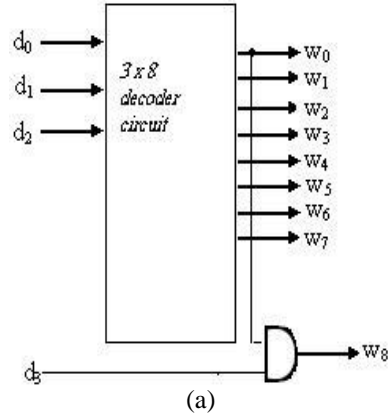
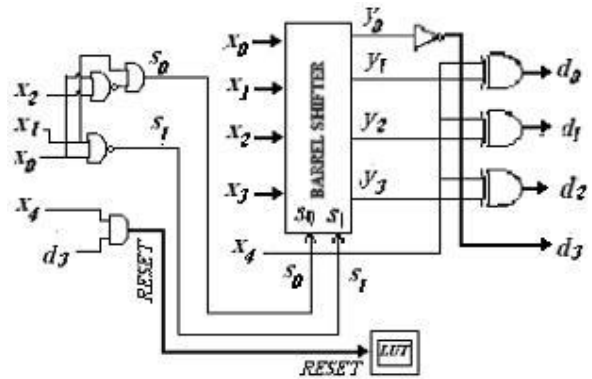


Figure 2. (a) Decoder circuit.



(b) Control circuit and Address generation circuit (AGC)

In addition to the shifter circuit, a memory unit for product values and a decoder circuit for mapping bits as well as a control circuit and address generation circuit are required. The mapping process of 5-bit input word to a 4-bit LUT address (d_0, d_1, d_2, d_3) is done by a simple set of mapping relations. The address bits are thus generated from the AGC as shown in figure 2(b) using the equation (3) and (4) that are defined below. Here the $Y_L \{y_0, y_1, y_2\}$ denotes all the shifted odd integer address bits. The relations used to map are as follows.

$$d_i = X_i'' + 1, \text{ for } i = 0, 1, 2, 3. \quad \dots (3)$$

$$d_3 = \overline{X_0''}$$

where $X'' = \{x_0'', x_1'', x_2'', x_3''\}$ is generated through address mapping the values after arithmetically right shifting the leading zeros of X' similar to that defined in equation

$$(1).i.e., X'' = Y_L, \text{ if } x_4 = 1; Y_L', \text{ if } x_4 = 0 \quad \dots (4)$$

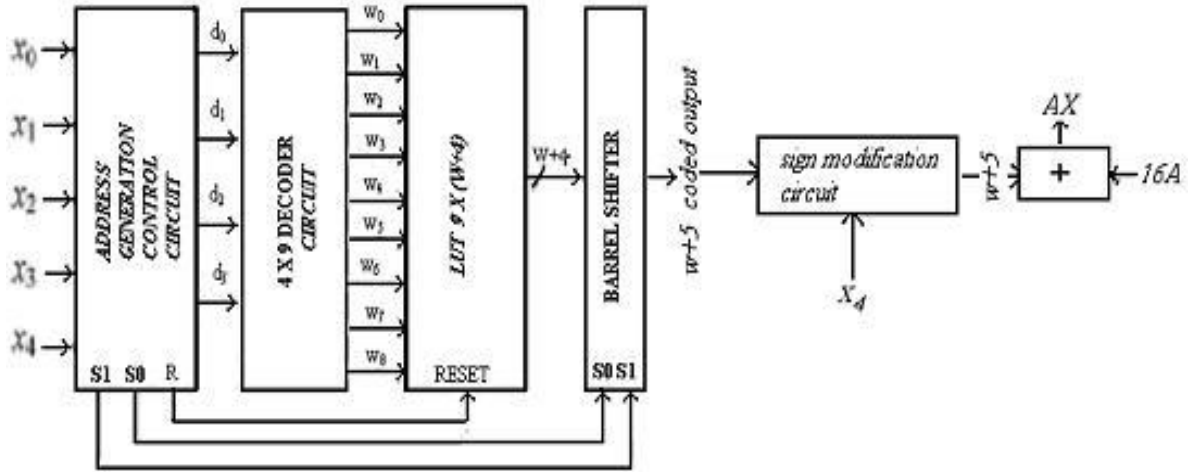


Figure 3. (a) A-OMS LUT System Block Diagram

For a given L-bit input word an address encoder maps the L-1 bit addresses of the LUT which consists of nine words of (W+4) –bit width. Modifying a simple 3 to 8 line decoder circuit (shown in figure 2(a)) produce a 4 to 9 line decoder that generates word select signals through which the required word from LUT is selected. In figure 2(b), a control circuit is shown to provide the control bits and a reset signal bit. The basic shift operation by barrel shifter is done by the control bits s₀, s₁ from control circuit. From the figure 2(b) the control bits s₀, s₁ are derived as follows.

$$\left. \begin{aligned} s_0 &= \overline{x_0 + (x_1 + \overline{x_2})} \\ s_1 &= \overline{(x_0 + x_1)} \end{aligned} \right\} \dots (5) \text{ and the reset signal is}$$

derived as

$$\text{RESET} = d_3 \text{ AND } x_4 \dots (6)$$

Where d₃ is defined in equation (3)

The optimized LUT circuit with modified schemes thus designed and is shown in figure 3. in figure 3 the address generation and control block is as shown in figure 2(b).

The main reason for approaching this technique is to optimize the implementation of the sign modification of the odd LUT output, which does not support the OMS scheme in methods defined previously [6]. The modified circuit is shown in figure 3. Also it provides the 2's complement representation of the product words, that supports computations with both the signed and unsigned bits, by modifying the (+/-)_cell (to perform add/subtract operations) of figure 1.

III. OPTIMIZED LUT BASED FIR SYSTEM DESIGN

The optimization of LUT using A -OMS method is clearly defined in the above section. As specified in the introduction the white spaces i.e. spectrum holes are detected using matched filters at the receiver end and FIR filter structure resembles the matched filter structure. Hence implementing an FIR filter and further designing a system based on the A-OMS method will be described here.

A. FIR Filter Design: A-OMS method is a different approach for implementing digital filters. The basic idea is to replace all multiplications and additions by a table & shifter-accumulator. An optimized FIR filter is designed, and the basic block diagram of the matched filter resembles the basic architecture of FIR filter. An FIR filter is a LTI digital filter that is characterized by the non-recursive difference equation in time domain and the equation is as follows

$$y[n] = \sum_{K=0}^{N-1} h[k]x[n-k] \dots (7) \text{ } x[n] = \text{Samples of input sequence,}$$

$$y[n] = \text{Samples of output sequences,}$$

$$h[k] = \text{Impulse response of the filter.}$$

The z-domain representation of it is as shown below

$$H(z) = Y(z) / X(z)$$

$$= \sum_{K=0}^{N-1} h[k] z^{-k} \dots (8)$$

To reduce the number of register and pipelined stages w.r.t the direct form, the transposed structure is considered. The direct form realization and Transposed form realization of FIR filter is show in figure 4(a), 4(b).

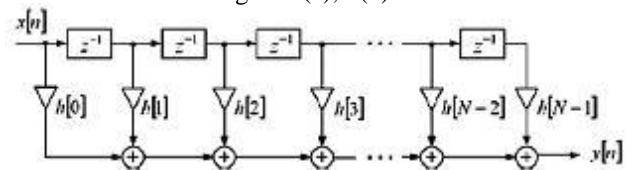
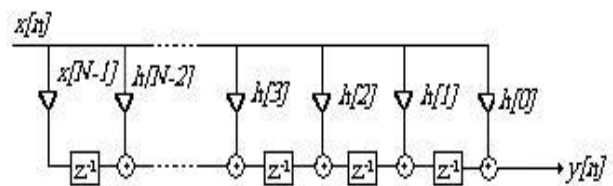


Figure 4. (a) Direct form realization of FIR filter



(b) Transposed structure realization of FIR filter

B. Design Process: FIR filter impulsive response is the ratio of output sequences to that of input sequences. A simple flow of FIR filter with A-OMS based design is shown in figure 5 where the input samples are represented by x[n] and multiplication operation by M, simple

arithmetic function by A, D stands for delay operator. The method specified in section 1 is applied to produce the desired output.

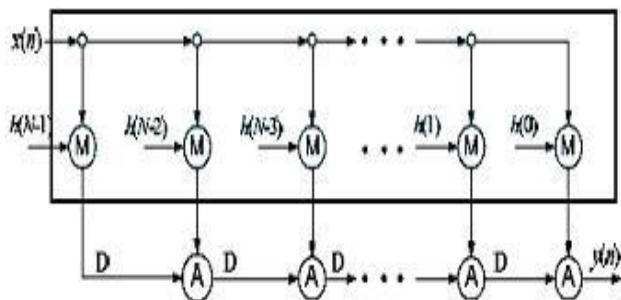


Figure. 5. A-OMS based FIR filter

The SA cells and AS cells are used to perform arithmetical operations add/subtract and logical operations arith-shift between the input samples and the impulses and to store them in a memory core that provide access through the LUT. One input sample in each clock cycle has the same number of cycles of latency as the optimized LUT since the same pair of address words are used by all the LUT-multipliers. The impulse sequences $h[0], h[1], h[2], \dots, h[N-2], h[N-1]$ are inputs of SA cell shown in figure 6.

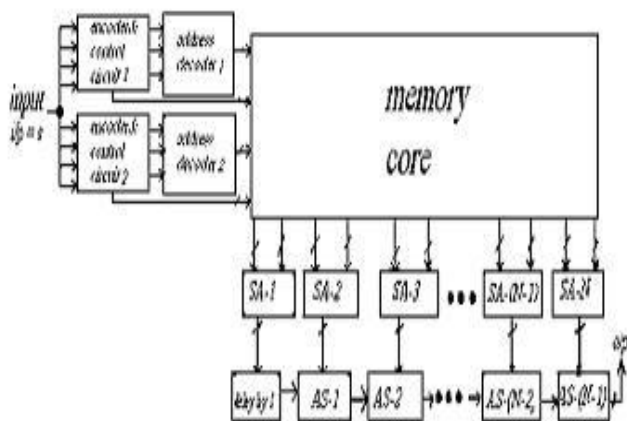


Figure. 6. FIR filter using optimized LUT structure

The decoder and control operations are similar to that described in section 2. The input values will be the input sequences $X[N]$ i.e., $(x[0], x[1], x[2], \dots, x[n-1])$ of the desired filter and the coefficient values are impulse sequences $h[0], h[1], h[2], \dots, h[N-2], h[N-1]$. The coefficient may be fixed or generic can be a set of impulse sequences. The A-OMS method can support multiple or a set of sequences and generic coefficient values along with fixed values. The desired output responses are derived through the mapping process that is explained in the section 2 of this paper. Finally A-OMS method reduces the computational delay.

The FIR filter design based on optimized LUT using A-OMS method provide more efficiency than the DA-based approach (that is done previously [2, 4]) in terms of area-complexity for a given throughput and lower latency of implementation. With an increase in the number of input sample size, the high precision multiplication

operations where input code words (length of code word) were coded into required word sizes and the A-OMS operations are performed in the same way as previous. This coding method is known to be the input coding technique that provides high precision operations.

IV. RESULTS

The comparisons in terms of the latencies and area complexity between the LUT-based design and the other methods based design are shown in table-1, table-2. FIR filter design using the High – Speed LUT structure thus implemented and simulated through XILINX ISE simulator and synthesis tools are used.

Table-1 Latencies for different filter order and input word size

N	LUT-Based Design			DA-Based Design		
	L = 8	L = 16	L = 32	L = 8	L = 16	L = 32
8	13	14	15	16	17	18
16	21	22	23	25	26	27
32	37	38	39	42	43	44
64	69	70	71	75	76	77
128	133	134	135	140	141	142

Table-2 Area complexity comparisons using different methods

FIR filter design	input operand width L	area (sq.µm)
DA-based	8	39029.56
	16	78895.96
conventional multiplier-based	8	40845.77
	16	82519.92
LUT -based	8	33880.80
	16	68558.92

The simulated results are as shown in the figure 7. The overall design process enhances the system performance in terms of speed and area that doubles the transmission rate, increasing the overall throughput. The result shows that more than 20% of saving in area-delay product with a transmission speed of twice that of the conventional methods.

It requires N times less number of decoders and memory requirement is reduced to 1/2 the of the conventional design therefore nearly 20% less area than conventional design methods (DA, Conventional Multiplier) for the implementation of a 16-tap FIR filter having the same throughput per cycle. This could be used for memory-based implementation of cyclic and linear convolutions, sinusoidal transforms, and inner-product computation. The results of the implemented FIR filter is also compared with the previous works shows reduction in the adder/subtract, slices, thus reduced memory core size.

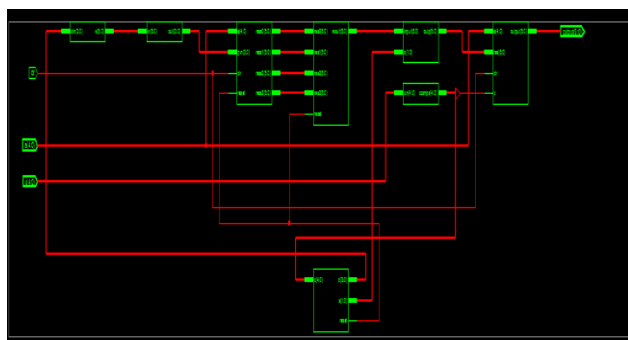
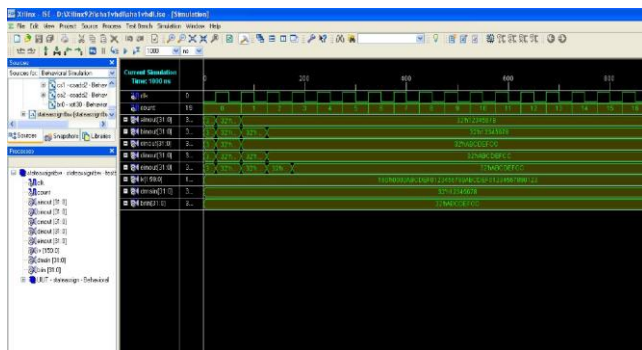


Fig. 7. Simulation Result and synthesis report for 32-bit

V. CONCLUSION

The proposed LUT multipliers for word size $L = W = 5$ and 6 bits are coded in Verilog and synthesized in XilinxISE 10.1i. Simulation Part is done in Modelsim 6.4b, where the LUTs are implemented as arrays of constants, and additions are implemented by the Wallace tree and ripple carry array. The CSD-based multipliers having the same addition schemes are also synthesized with the same technology library. we have shown the possibility of using LUT based multipliers to implement the constant multiplication for DSP applications.

REFERENCES

- (1) Verilog hdl by padnabam for velog language.
- (2) LUT Optimization for Memory-Based Computation by. Pramod Kumar Meher, *Senior Member, IEEE*
- (3) P.K.Meher, 'LUTOptimizationforMemory-Based computation,'IEEE Trans on Circuits & Systems-II, pp.285-289, April 2010.
- (4) Professor S.K. Sanyal, Wasim Arif, "Designing of a fast LUT based DDA FIR system with adaptive coefficient for Spectrum Sensing in Cognitive Radio" ICGST AIML-11 Conference, Dubai, UAE, 12-14 April 2011.
- (5) P. K. Meher, 'New Approach to Look-up-Table Design and Memory-Based Realization of FIR Digital Filter,' IEEE Trans on Circuits & Systems-I pp 592-Systems I, pp.592 603, March 2010.
- (6) Tevfik Y'ucek and H'useyin Arslan, "A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications", IEEE Communications Surveys & Tutorials, Vol. 11, No. 1, First Quarter 2009
- (7) H. H. Dam, A. Cantoni, K. L. Teo, and S. Nordholm, "FIR variable digital filter with signed power-of-two coefficients," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 6, pp. 1348-1357, Jun. 2007.
- (8) Verilog hdl by padnabam for velog language.
- (9) LUT Optimization for Memory-Based Computation by. Pramod Kumar Meher, *Senior Member, IEEE*
- (10) P.K.Meher, 'LUTOptimizationforMemory-Based computation,'IEEE Trans on Circuits & Systems-II, pp.285-289, April 2010.
- (11) Professor S.K. Sanyal, Wasim Arif, "Designing of a fast LUT based DDA FIR system with adaptive coefficient for Spectrum Sensing in Cognitive Radio" ICGST AIML-11 Conference, Dubai, UAE, 12-14 April 2011.
- (12) P. K. Meher, 'New Approach to Look-up-Table Design and Memory-Based Realization of FIR Digital Filter,' IEEE Trans on Circuits & Systems-I pp 592-Systems I, pp.592 603, March 2010.
- (13) Tevfik Y'ucek and H'useyin Arslan, "A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications", IEEE Communications Surveys & Tutorials, Vol. 11, No. 1, First Quarter 2009
- (14) H. H. Dam, A. Cantoni, K. L. Teo, and S. Nordholm, "FIR variable digital filter with signed power-of-two coefficients," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 6, pp. 1348-1357, Jun. 2007.