

Entertainment Systems and High-Performance Processor SH-4

Norio Nakagawa
Fumio Arakawa

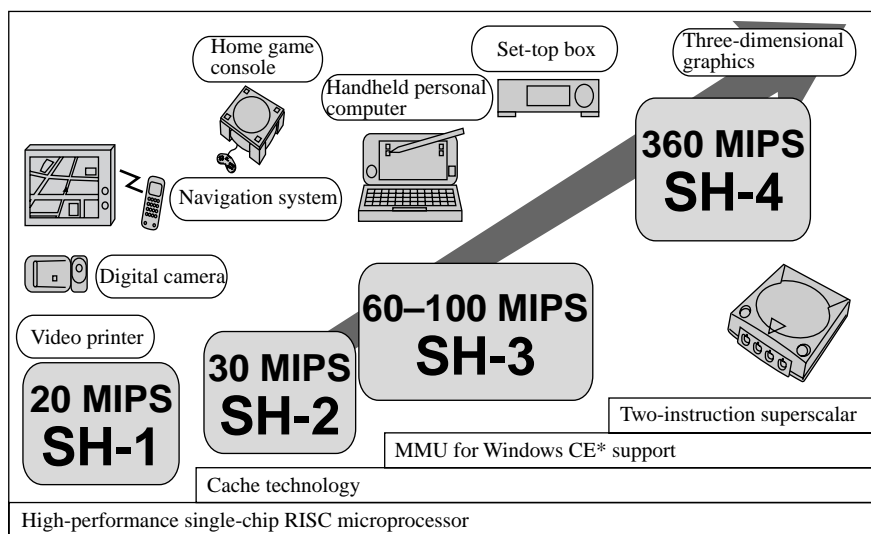
*OVERVIEW: Sega Enterprises, Ltd. announced its next-generation high-performance entertainment console Dreamcast*¹ at a press conference on July 20, 1998 and won widespread acclaim during the demonstration of its powerful and precise video images. Dreamcast's nucleus is Hitachi, Ltd.'s SH-4 microprocessor. SH-4 is the top-of-the-line model in Hitachi's high-performance SuperH series reduced instruction set computer (RISC) microprocessor. It was developed for next-generation multimedia appliances and specifications were tuned-up based on performance in game consoles. This was done because we believe that game consoles contain all the elements that will be needed in products for the multimedia era. Three elements especially were considered when developing the SH-4: support of middleware by enhancing the performance of the central processing unit (CPU); necessity of three-dimensional graphics support for communications; and enhancement of data transfer performance for data types such as video and audio. Specifications achieved include integer performance of 360 million instructions per second for the SH-4 when operated with an internal clock of 200 MHz, image processing performance of 5 Mpolygons/s, and a maximum data transfer rate of 800 Mbyte/s.*

INTRODUCTION

WITH the arrival of the real multimedia era there is a demand for appliances that are cheaper, feature higher performance, and can be used for multiple purposes. The processor used in the implementation of such appliances must have greatly improved performance. The following factors provide the reasons.

(1) Smaller and less-expensive systems can be developed by implementing in software communications and other facilities that previously were executed by dedicated hardware. Furthermore, there is demand for advanced appliances with the capability

*¹ Dreamcast is a registered trademark of Sega Enterprises, Ltd.



MIPS: million instructions per second
MMU: memory management unit
RISC: reduced instruction set computer

Fig. 1—SuperH Series High-Performance Single Chip RISC Microprocessor Application Fields. SuperH series top-of-the line SH-4 supports enhanced speed in three-dimensional graphics for use in high-performance amusement systems.

* Windows is a registered trademark of Microsoft Corp. of the US in the US and other countries.

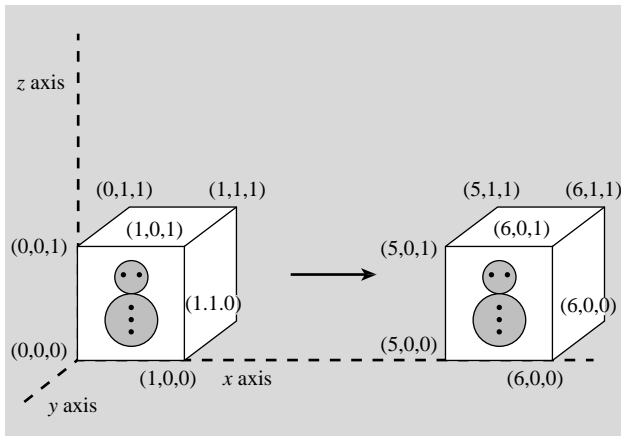


Fig. 2—Display of a Regular Hexahedron .
When a regular hexahedron is displayed using quadrilateral polygons, it can be represented by 8 coordinates.

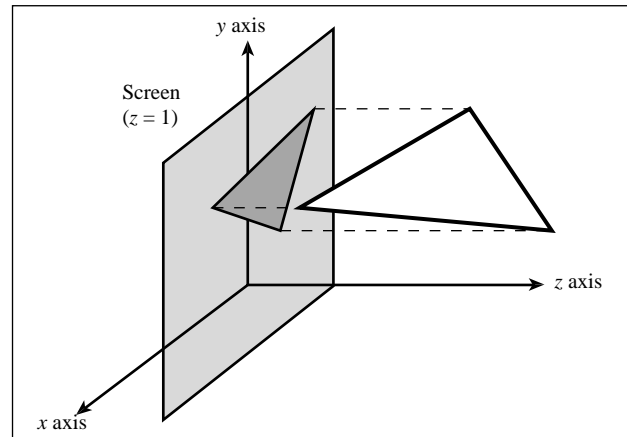


Fig. 3—Projection of a Polygon.
Three-dimensional polygon is projected and drawn on screen at $z = 1$.

to create highly flexible systems that can be modified or updated to later versions by reloading software in the same manner as ordinary personal computers.

(2) To further expand the multimedia appliance market, it is necessary to market these appliances not only to persons who utilize personal computers but to a much broader user population. Thus a worldwide common user interface is desired that is user friendly. A higher-level user interface should be implemented that includes upgraded image display quality, handwriting input, and voice recognition.

Hitachi confirmed that its RISC microcomputer SH-4 meets these requirements by using a game console as the basis for adding various facilities and their review. Here we will discuss how the floating-point inner-product execution unit added to the SH-4 enhances the speed of three-dimensional graphics.

THREE-DIMENSIONAL GRAPHICS OVERVIEW

Recently, three-dimensional graphics have entered the main stream in game consoles as well as in the motion picture and scientific fields. High-performance three-dimensional graphics have thus become indispensable in next-generation high-performance systems.

During image processing, the processor performs coordinate and other computations in the geometric mode, and the graphic chip performs color placement and other computations in the drawing mode. We will discuss the geometric mode below.

In three-dimensional graphics, the surfaces of all objects are overlaid with triangles called polygons (quadrilateral polygons are sometimes used), and the

three-dimensional coordinates of their vertices (x , y , z) are used to represent the image. Also, by transforming these vertices, movement can be represented.

Fig. 2 shows the movement of a regular hexahedron. If quadrilateral polygons are used to represent the hexahedron, it can be represented by six polygons and eight vertices. To move this regular hexahedron five units parallel to the x axis, all vertices are transformed by adding the value 5 to their x coordinate. The vertices after the movement can be computed in this manner.

The polygon is projected to display it on the screen, as shown in Fig. 3. To speed up the process, we can assume that the screen is at the position $z = 1$. This procedure is assigned to the graphic chip to execute the drawing process. When the CPU is used for graphics, a method for rapidly executing this coordinate transformation and projection must be programmed.

HIGH-PERFORMANCE AMUSEMENT SYSTEMS AND PERFORMANCE REQUIREMENTS

The performance requirements for high-performance amusement systems are extremely high and exceed the level formerly considered reasonable. The following two reasons can be given.

Increase in number of polygons

Fig. 4 shows an example of a sphere represented by polygons. In Fig. 4 (a) the sphere is represented as a regular octahedron, but this solid will not be recognized as a sphere. However, when the number

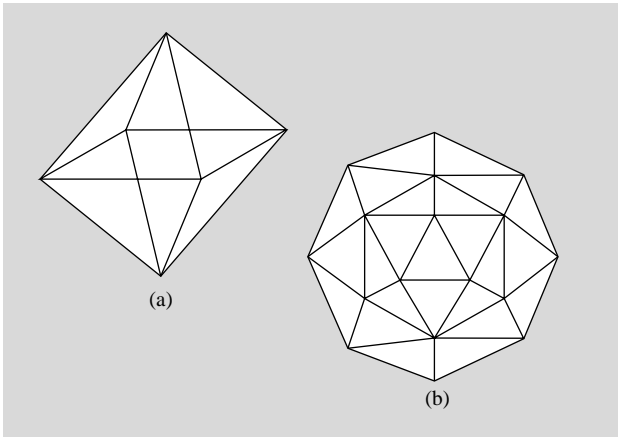


Fig. 4—Sphere Representation and Number of Polygons. The representation of a sphere appears smoother as the number of polygons used to represent it increases.

of triangular polygons is increased as in Fig. 4 (b), the object becomes closer in appearance to a sphere. Thus, in a high-performance amusement system, finer-grain polygons are used for a more precise representation, with the aim of providing a high-definition display.

However when the number of triangular polygons is large, the number of vertexes obviously increases, and a greater amount of data are required. Moreover, the number of coordinates required to move these objects increases.

Increase in transformation count

It is not sufficient to represent each individual object in detail; a representation even closer to the real thing is required. Fig. 5 shows an example of the polygons representing a hand. Up until now, a hand would be shown like a stick in distant views. In near views of the hand only, the screen would be changed to a detailed drawing mode with an accurate model representing the hand by clusters and joints.

A recent trend, though, is for software to provide the capability of freely zooming in and zooming out as a matter of course. Also game software development has progressed to the large-scale project size, and it is no longer realistic to create a model for each scene. Because each object is represented by a single model, everything must be represented by a detailed model. Thus it is necessary to compute everything from a person’s location to the coordinates of a fingertip every time, and not only the transformed coordinates but also the number of transformations themselves has increased.

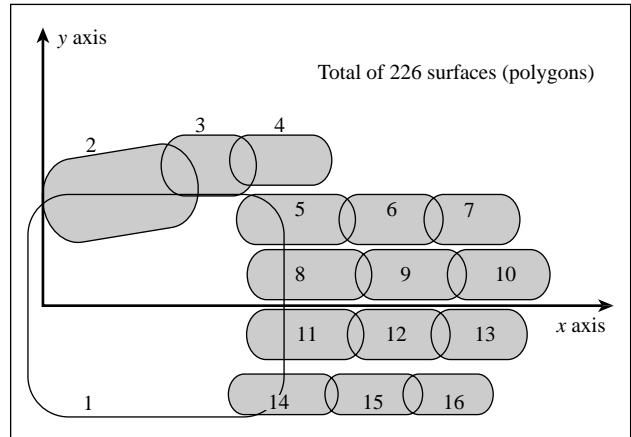


Fig. 5—Polygon Model of a Hand. Accurate representation of a hand including joints requires 16 clusters and 226 polygon surfaces.

Dynamic computation introduced

Images presented with higher precision must be moved in a smoother manner. Motion, that is the mode in which an object moves along the time axis, is not defined at present by any theory. Thus it is determined by actually moving and observing a model. However, in the future we expect to be able to compute motion accurately by following natural laws such as those relating to gravitational force and muscular strength, and by using these computational results depict even more natural motion. We want to emphasize, though, that to make such a presentation means that even more computation will be required.

SOLUTIONS ON THE SH-4

We deliberated on methods of supporting solutions to the above problems using a general-purpose processor. It is possible to implement all of these functions on a general-purpose processor, but difficulties would arise with respect to cost and development time. Thus we decided to add support for two functions, floating-point execution support and matrix computation execution unit support.

Floating-Point

We will now discuss the necessity of providing floating-point support.

CPU performance restrictions

The Super-H series uses 16-bit instruction-length codes to increase the instruction-code efficiency. Thus it has only 16 general-purpose registers, and moreover

it does not have many unused instruction codes. To add additional facilities, it is necessary to steal performance from the CPU, or create another execution unit and transfer data through the CPU registers. These methods will definitely restrict the performance of the CPU itself. Therefore, to add facilities it is necessary to support a processor type such as a floating-point processor or a digital signal processor (DSP).

Magnitude of the absolute representation range

With floating-point it is possible to represent more than 70 digits in the decimal system, but with fixed-point decimals the range that can be represented in the decimal system is only about 10 digits. Thus this limitation of a fixed decimal point is likely to be a bottleneck to general purpose use in a wide variety of multimedia systems.

When smooth motion is executed with dynamic computation to produce a realistic feeling, it is obvious that jumps or dives should not be shown as uniform-velocity motion. Instead it is necessary to compute the position of each object for accelerated motion due to gravity. With fixed-point decimals, it is necessary to continually be aware of the number of effective digits when developing programs including computation of momentum having directionality such as collisions or rebounds — or for even more complex computations, with stickiness or air resistance.

For example, if a 10,000-t rocket travels 100 m/s then the momentum is 1,000,000,000 kg·m/s, but if a 10-mg hair moves 5 cm/s then it is 0.0005 kg·m/s. When we work with both the background and foreground of a three-dimensional graphic simultaneously, there is a difference of more than 13 decimal places, and we cannot fully display such an

image with fixed-point decimal. Of course, a programmer who accurately controls the decimal places can successfully dodge this problem, but it requires an excessive amount of labor.

Maintaining relative precision

If we take an image as an example, and have a word length of 32 bits, we can represent distances from 1 mm to 2,000 km in integer notation. This provides adequate representation power for a still picture, but if one wants to represent moving objects in full-motion video, then cumulative rounding errors become a problem. That is, if the least significant digit is always rounded off, then an error will always occur with a probability of 1/2. Even if the precision of the smallest value represented is 1 mm, after 100 operations (computations), a maximum error of 50 mm may occur.

If the image is viewed from a distance of 1,000 km there is no problem, but a displacement of 50 mm viewed from about 1 m will make a big difference in the impression of the pattern. For example, a hand may appear to be detached from the body; conversely, it may appear to be embedded in the body.

At present, image software incorporates the complexity of relatively detailed schemes to compensate for this effect and for scale adjustment, etc. With floating point, though, the effective number of digits is always maintained so that relative error is constant, and this type of processing is basically not needed. Thus we decided to incorporate floating-point support.

Matrix Processing Support

Coordinate transformations in three-dimensional geometry are in general called affine (pseudo) transformations that are represented by a 3×4

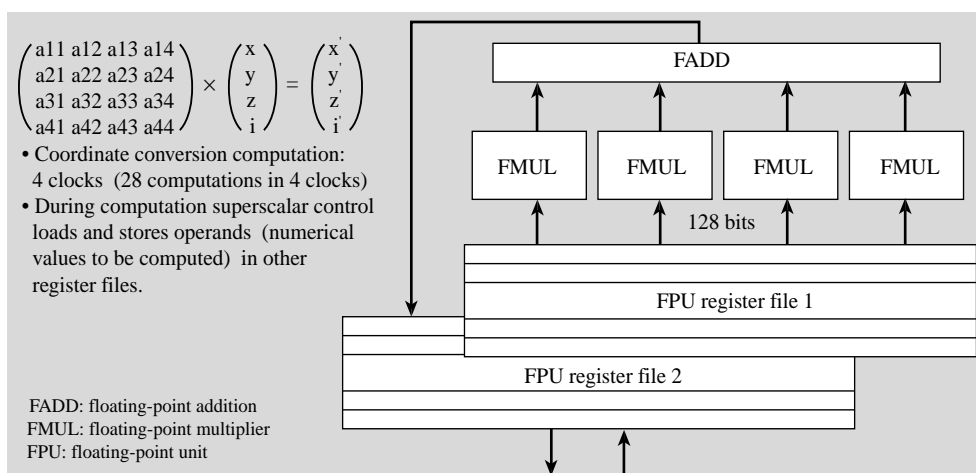


Fig. 6—Floating-Decimal Point Four-Dimensional Inner-Product Execution Unit.

We added hardware facilities that can execute the inner product of four-dimensional vectors in a single clock. This implementation provides high-speed execution of affine and other conversions, and fully satisfies the performance requirements.

determinant, or the parallel translation component may be eliminated for implementation by a 3×3 matrix. Since deformation may be added, though, formal graphics transformation requires a 4×4 matrix.

We considered the recent rapid progress in graphics processing technology and implemented a 4×4 matrix execution unit in the SH-4 to support deformation. This execution unit can be used for all applications including dynamical systems and electromagnetic wave systems as well as for image processing.

Instructions support the operations shown in Fig. 6 for the reasons given above. That is, the determinant is 4×4 , but affine transformations can be executed by issuing 3 instructions. Moreover, execution-time overhead is unchanged from support of 3×3 only because parallel translation computation can be done at the same time.

Future speed-up of graphics transformations will not be executed by individually transforming the coordinates of each coordinate. Rather a method is needed in which Eulerian angles and a central coordinate system are introduced, the transformation matrix itself processed in advance, and in the final step the vector is transformed in a single operation. This type of operation is not impossible with 3×3 , but the algorithm would become complex so it was decided that at least a 4×3 matrix is necessary.

THREE-DIMENSIONAL GRAPHICS MEASUREMENT RESULT

We wrote a program using a simple model to ascertain the actual three-dimensional graphics performance, and performed a test.

Model structure

As shown in Fig. 5, we used a model of a hand as a simple model. This model of a hand consists of a palm and five fingers, which are divided into sections by joints. We separated the individual clusters to obtain a structure with a total of 16 clusters. Multiple polygons are used to represent the clusters, with 30 polygons used for the palm, and 12 or 13 polygons used for the other clusters, for a total of 226 polygon surfaces.

Program outline

Procedures for operations that move the five fingers as described above are shown in Fig. 7.

- (a) move (): Multiply the present matrix by the parallel movement matrix
- (b) rotate Y (): Multiply the present matrix by the y

```

/* C: cluster connections: numbers correspond to diagram
C1-C2-C3-C4
-C5-C6-C7
-C8-C9-C10
-C11-C12-C13
-C14-C15-C16*/
move (C1, optional position); rotate(C1, optional rotation);
                                coordinate conversion (C1)
move (C2 →1); rotate Y (C2, 1 degree);
                                coordinate conversion (C2)
move (C3 →2); rotate Y (C3, 1 degree);
                                coordinate conversion(C3)
move (C4 →3); rotate Y (C4, 1 degree);
                                coordinate conversion (C4)
move (C5 →4); rotate Y (C5, 1 degree); coordinate conversion (C5)
move (C6 →5); rotate Y (C6, 1 degree);
                                coordinate conversion( C6)
move (C7 →6); rotate Y (C7, 1 degree);
                                coordinate conversion( C7)
    
```

Fig. 7—Conversion Computation Order.

- axis rotational matrix
- (c) rotate (): Multiply the present matrix by the optional rotational axis designated by Eulerian angle, etc.
- (d) coordinate transformation: Have object inside cluster act on present matrix

Project the coordinates determined in the above manner, do brightness and other necessary computations, and display

Measurement results

Measurements of SH-4 performance made using the above program show that it realizes a geometric execution performance of 5 Mpolygons/s. The ratio of performance using an inner product and not using an inner product is shown in Fig. 8.

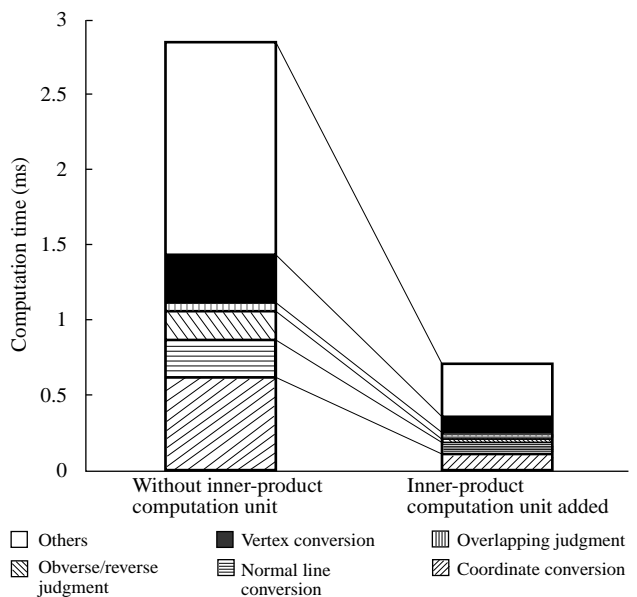


Fig. 8—Effect of Inner-Product Computation Unit. Incorporation of an inner-product computation unit implements speed up of three-dimensional graphics.

TABLE 1. SH-4 Chip Specifications

Performance emphasized rather than low power consumption.

Operation frequency	Internal: 200 MHz, external: 100 MHz	FLOPS: floating-point operations per second
Performance	Integer section: 360 MIPS at 200 MHz, 300 MIPS at 167 MHz	SDRAM: synchronous dynamic random access memory
	Floating-Point Section: 1.4 GFLOPS at 200 MHz, 1.12 GFLOPS at 167 MHz (Peak values)	EDO: extended data output
Superscalar	Simultaneous execution of two instructions	SRAM: static RAM
Cache capacity	8 kbyte (instructions), 16 kbyte (data)	DMAC: direct memory access controller
Memory types supported	Direct connection interface for SDRAM, DRAM (EDO, hyperpage), and SRAM	SCI: serial communication interface
Maximum data transfer rate	800 Mbyte/s	SCIF: serial communication interface with FIFO
Peripheral modules	4-channel DMAC, timer, SCI/SCIF, and RTC	RTC: real-time clock
Power consumption	Typ. 1.5 W at 200 MHz	LM: layer metal
Process	0.20 μm , 5 LM, 1.8 V, with 3.3-V I/O	I/O: input and output
Package	256 BGA or 208 QFP, 64-bit data bus	BGA: ball grid array
		QFP: quad flat package

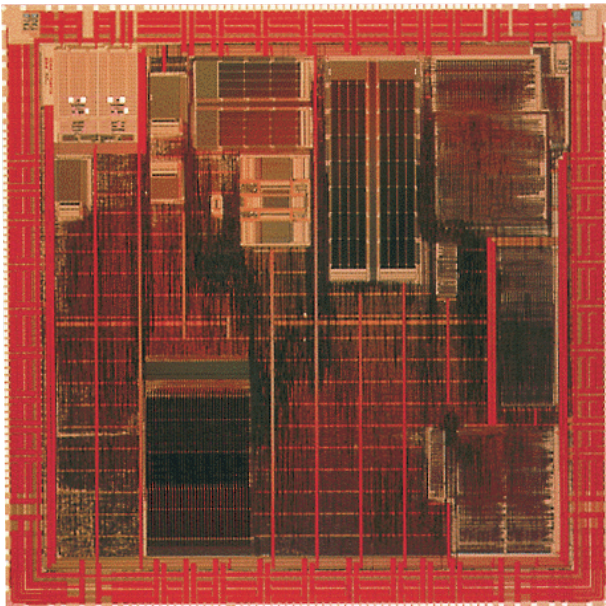


Fig. 9—SH-4 Chip.

The chip is the key to three-dimensional graphics.

CONCLUSIONS

We have described how Hitachi's SH-4 single-chip RISC microprocessor achieves a 3-to-4 times speed-up in three-dimensional graphics processing compared with earlier chips. Various means have been employed to speed-up high-performance amusement appliances, including higher-speed graphics, a large improvement in the data transfer rate, and reduction in cache-control overhead.

Fig. 9 shows a photo of the chip, while its

specifications are shown in Table 1. As can be seen from Table 1, the SH-4 has been designed for high performance rather than low power consumption. However we are now developing a low-voltage low-power consumption version using the same basic architecture for mobile information appliances running Windows CE and other OSs by circuit and semiconductor device tuning.

REFERENCES

- (1) F. Arakawa et al., SH-4 RISC Microprocessor for Multimedia, Hot CHIPS, IX (1997-8), pp. 165-176
- (3) O. Nishii et al., A 200 MHz 1.2W 1.4 GFLOPS Microprocessor with Graphic Operation Unit, Int. Solid-State Circ. Conf. (1998-2)
- (4) F. Arakawa et al., SH-4 RISC Multimedia Microprocessor, IEEE MICRO Chips, Systems, Software, and Application (1998-3, 4)

ABOUT THE AUTHORS



Norio Nakagawa

Joined Hitachi, Ltd. in 1983, and now works at the SH Design Dept. of Semiconductor & Integrated Circuits Div. He is currently engaged in design and development of microprocessors. Mr. Nakagawa can be reached by e-mail at nakagawa@cm.musashi.hitachi.co.jp.



Fumio Arakawa

Joined Hitachi, Ltd. in 1986, and now works at the System LSI Research Dept. of Central Research Laboratory. He is currently engaged in design and development of microprocessors. Mr. Arakawa can be reached by e-mail at arakawa@crl.hitachi.co.jp.