объединенный
институт
ядерных
исследований
дубна

E10-82-793

A.Leich, A.D.Polyntsev

THE IMPLEMENTATION OF TWO-DIMENSIONAL TRANSFORMATIONS ON THE ARITHMETICS MODULE OF AN INTELLIGENT GRAPHICS TERMINAL

Submitted to the Conference on Computer Graphics´83 (ČSSR, 1983)

1982

1. ARCHITECTURE OF THE ARITHMETICS MODULE

The arithmetics module [LEI82] contains two special arithmetic processing units (APU), which provide fixed and floating point arithmetic and a variety of floating point trigonometric and mathematical operations. The APUs (Am9511) have no means to fetch operands and operation codes, therefore these functions are performed by a control processor (CPU) - INTEL 8080. The arithmetics module is connected with the IGT's [LEI81] common memory via the common bus.

The arithmetics module's local memory consists of 8K byte PROM and a 2K byte RAM. A part of the RAM, the control memory, may be accessed from the common bus of the IGT as well as from the CPU of the arithmetics module. The control memory is used as a mailbox for messages to be transferred to/from the arithmetics module.

2. PROGRAMMING THE ARITHMETIC PROCESSING UNIT

Prior to performing an operation on the Am9511 operands have to be loaded (pushed) into its internal stack memory. A command issued from the CPU causes operations on the data in the stack to be performed. The result is then available to be retrieved from the stack, or additional data or commands may be entered. All transfers including operand, result, status and command information take place over an 8-bit data bus.

Some experience from applications of the Am9511 illustrating how to use the Am9511's internal stack for storing intermediate results and the employment of two APUs for simultaneously computing symmetric expressions is given in [STAU]. It was shown, that using two APUs instead of one may lead to a speed-up of 36%.

3. THE TRANSFORM SUBROUTINE

The main computations necessary for two-dimensional transformations [NEW] are given by:

$$x' = a*x + b*y + c,$$

$$y' = d*x + e*y + f.$$

As these calculations are applied to a set of world coordinate pairs (x,y) it is important to perform them by means of an optimal subroutine in terms of execution time. The values of a,b,c,d,e and f are to be defined in the arithmetics module in accordance with transformation parameters like displacement values, scaling factors, rotating angle, window and viewpoint boundaries only once for a desired transformation and are available in floating point format. The coordinates may be given in one of the following formats: 16- or 32-bit fixed point or 32-bit floating point.

The simplified algorithm of the subroutine for performing these computations on coordinates in floating point format using two APUs is shown below:

STEP	OPERATION	STACK OF 1st APU	STACK OF 2nd APU
1	push x to 1 and 2	x	x
2	push a to 1	a; x	x
3	push d to 2	a; x	d; x
4	multiply 1 and 2	a*x	d*x
5	push y to 1 and 2	y; a*x	y; d*x
6	push b to 1	b; y; a*x	y; d*x
7	push e to 2	b; y; a*x	e; y; d*x
8	multiply 1 and 2	b*y; a*x	e*y; d*x
9	add 1 and 2	b*y + a*x	e*y + d*x
10	push c to 1	c; b*y + a*x	e*y + d*x
11	push t to 2	c; b*y + a*x	f; e*y + d*x
12	add 1 and 2	x´	y´

In this subroutine the push operations represent subroutine CALLs to fetch 4-byte operands from the memory and push it onto the APU´s stacks. The operations multiply and add represent operation code transfer (1 byte) to the APUs and performing of floating point arithmetic operations in the APUs.

The push operations in step 5 and 10 may be performed partially in parallel to step 4 and 9 if the whole operand is initially fetched from the memory into the CPU´s internal registers and transferred to the APU afterwards. This leads to a speedup of 18% for the transform subroutine.

During steps 4,5,9 and 10 both APUs operate in parallel. This contributes a gain of 35.6% of execution time assuming worst case values for operand depending execution time in the APUs. The average execution time of the subroutine is 780 microseconds with a clock frequency of 2MHz for CPU and APUs.

Similar subroutines perform the same computations on input data given in the 16- or 32-bit fixed point format. They contain an additional step for format conversion of the source data x and y after steps 1 and 5. For 16-bit operands the push subroutine will transfer only two bytes.

4. CLIPPING AND TRANSFORMATION

No pop operations for retrieving the results are required for the described transform subroutine, as a clipping algorithm may be performed after the transformation. In this case the output values of the transform subroutine are kept in the APU´s stacks as input variables for the clipping subroutine. The clipping algorithm for single points merits from the concept of two APUs too. The algorithm for vector clipping follows the description given in [NEW] and does not contain symmetric expressions for the x- and y-components of a coordinate pair.

Clipping is not required if the range of coordinate values to be transformed (RANGE) lies within the window boundaries. If RANGE is given as input parameter, the arithmetics module determines after a comparison of WINDOW and RANGE whether clipping has to be performed or not.

An additional analysis is performed for points whether the transformation contains a rotation or not. If no rotation is required then clipping occurs before transformation and the transform operations for points outside the window have not to be performed.

5. INTERFACE

The array of world coordinates to be transformed as well as the result coordinates are to be stored in the common memory of the IGT. Buffer address, length and format for input data (x,y) as well as for output data (x´,y´) and also the transfotmation parameters are specified in the mailbox. Thus it is possible to use the arithmetics module also in other configurations independently from the IGT.

6. CONCLUDING REMARKS

The algorithms for transformation and clipping have been tested at the initial stage by means of writing a FORTRAN program on INTELLEC MDS. After the preliminary evaluation the program has been rewritten in INTEL 8080 Assembly language, thus increasing program performance in the given hardware environment of the arithmetics module. The final version of the program has a length of about 5K byte.

REFERENCES

[LEI81] Leich H., Levchanovsky. The Structure of a Multi-Microprocessor System for an Intelligent Graphic Terminal. JINR, E11-81-297, Dubna, 1981.

[LEI82] Leich H. The Architecture of an Arithmetics Module for the Intelligent Graphics Terminal. To appear in: Almanac of Conference on Computer Graphics´83. Bratislava, 1983.

[NEW] Newman W.M., Sproull R.F. Principles of Interactive Computer Graphics. McGraw-Hill, 1973.

[STAU] Stauffer M.K. Math.Processor Chips Boost μC Computing Power. EDN, August 20, 1980, pp.113-120.

Received by Publishing Department
on November 22,1982.

Лайх А., Полынцев А.Д. E10-82-793
Реализация двумерных преобразований арифметическим модулем интеллектуального графического терминала

Мультимикропроцессорный интеллектуальный графический терминал /ИГТ/, разработанный в ОИЯИ, Дубна, автономно выполняет двумерные преобразования мировых координат. Операции типа смещение, масштабирование, вращение, преобразования для выделения окна данных и окна экрана осуществляются посредством специального арифметического модуля ИГТ. В представленной статье обсуждаются некоторые аспекты использования трехпроцессорной архитектуры для параллельной обработки мировых координат.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1982

Leich A., Polyntsev A.D. E10-82-793
The Implementation of Two-Dimensional Transformations on the Arithmetics Module of an Intelligent Graphics Terminal

The multi-microprocessor based Intelligent Graphics Terminal - IGT developed at JINR, Dubna, autonomously performs two-dimensional transformations of world coordinates. Operations like displacement, scaling, rotation, window/viewport transformation and clipping are implemented on a dedicated part of the IGT, the microprocessor based arithmetics module. This paper outlines some topics of using a three processors architecture for parallel processing of world coordinate pairs.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Preprint of the Joint Institute for Nuclear Research. Dubna 1982