

ПАРАЛЛЕЛЬНЫЕ АППАРАТНЫЕ СРЕДСТВА ГЕНЕРАЦИИ ДИНАМИЧЕСКИХ ИЗОБРАЖЕНИЙ

Авксентьева О.А., Викторов Е.В.

Кафедра ЭВМ ДОНГУ

E-mail: avksen@cs.dgtu.donetsk.ua

Abstract

Avksentyeva O., Victorov E. Parallel hardware of generation of the dynamic images. The article is devoted to development and research of a parallel mathematical model of affine mappings and structure of device on the basis of technology of transputers.

Ставшая классической структура системы синтеза изображений основана на конвейере вычислителей, функционально ориентированных на реализацию алгоритмов синтеза [1]. Система включает: базу данных об отображаемой сцене; сценарный процессор, реализующий принятый алгоритм формирования дисплейного списка; геометрическую подсистему, выполняющую необходимые геометрические преобразования (поворот/смещение векторов, масштабирование, отсечение, проецирование и др.); подсистему растривания, производящую растровое разложение графических примитивов; устройство отображения.

Основополагающими принципами построения современной графической системы служат: аппроксимация поверхностей объектов и иерархия моделей объектов, параллельные вычисления и организация вычислительного конвейера, применение эффективных алгоритмов, реализуемых специализированными вычислителями. Применение специализированных вычислителей, работающих параллельно, увеличивает мощность системы при удовлетворении требованиям режима реального времени (образ объекта или сцены на экране должен перевычисляться и обновляться с частотой порядка 30 кадров/с).

Аффинные преобразования находят широкое применение при решении задач машинной графики [1]. Наибольшее распространение получили частные случаи аффинных преобразований: сдвиг, масштабирование, вращение и преобразования, полученные их последовательным применением.

Преобразование координат выполняется посредством умножения обобщенной матрицы коэффициентов A на исходный вектор координат $b = (x \ y \ z \ 1)$. В результате получим вектор преобразованных координат $c = (x' \ y' \ z' \ 1)$.

$$c = Ab = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & 1 \end{pmatrix} * \begin{pmatrix} x & y & z & 1 \end{pmatrix} = \begin{pmatrix} x' & y' & z' & 1 \end{pmatrix}$$

Координаты x' , y' , z' получаем в соответствии с формулами:

$$\begin{aligned} x' &= a_{11} * x + a_{21} * y + a_{31} * z + a_{41} \\ y' &= a_{21} * x + a_{22} * y + a_{32} * z + a_{42} \\ z' &= a_{31} * x + a_{32} * y + a_{33} * z + a_{43} \end{aligned} \quad (1)$$

Схема алгоритма последовательного вычисления координат x' , y' , z' приведена на рис. 1. Во внешнем цикле по переменной i выбирается координата для преобразования $c[i]$. Во внутреннем цикле по переменной j мы производим умножение соответствующих коэффициентов матрицы A на соответствующие координаты $b[j]$ и суммирование полученного

произведения с ранее полученной частичной суммой $c[i]$. В результате получаем преобразованные координаты. Таким образом, нам необходимо произвести 9 операций умножения и 9 операций суммирований чисел с плавающей запятой.

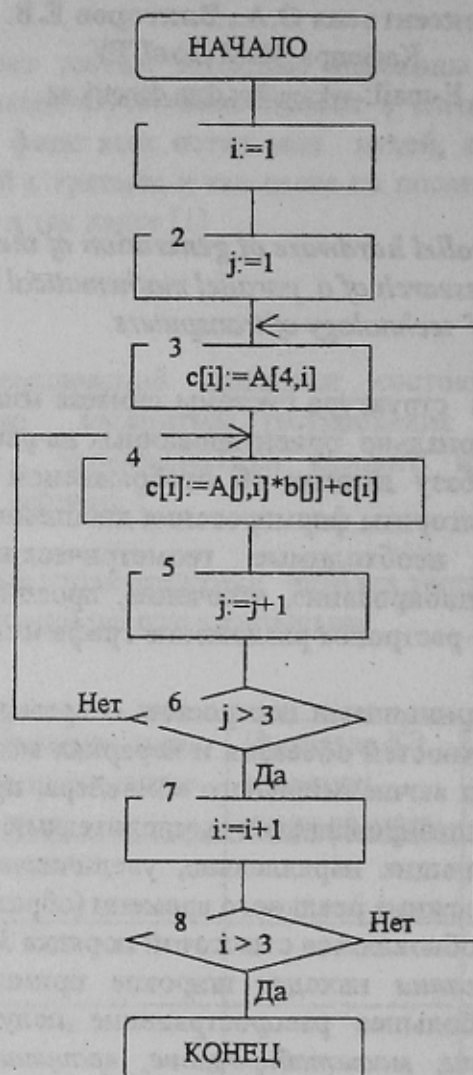


Рис. 1 – Последовательный алгоритм преобразования

Как можно заключить из формул (1), нахождение преобразованных координат x' , y' , z' можно производить отдельно друг от друга, в силу их взаимной независимости [2]. Исходя из последовательного алгоритма преобразования, видим, что можно выполнять параллельно умножение координат на коэффициенты, а так же сложение полученных парных произведений для всех искомых координат одновременно. Схема, иллюстрирующая параллельное вычисление координаты вектора, приведена на рис. 2. На первом уровне выполняется произведение исходных координат на элементы множеств a_i . Множества a_i являются строками обобщенной матрицы преобразования A . На втором и третьем уровне реализуется метод сдвигания для суммирования парных произведений, переданных с 1-го уровня. На выходе схемы получаем элемент вектора преобразованных координат $c = (x' y' z' 1)$. Таким образом, для вычисления всех элементов вектора c нам необходимо произвести 3 операции умножения и 6 операций суммирований.

Видно, что реальное параллельное выполнение аффинных преобразований невозможно производить на однопроцессорной ЭВМ, для этого необходимо несколько процессорных элементов. В статье определяются оценки параллельных архитектур для реализации

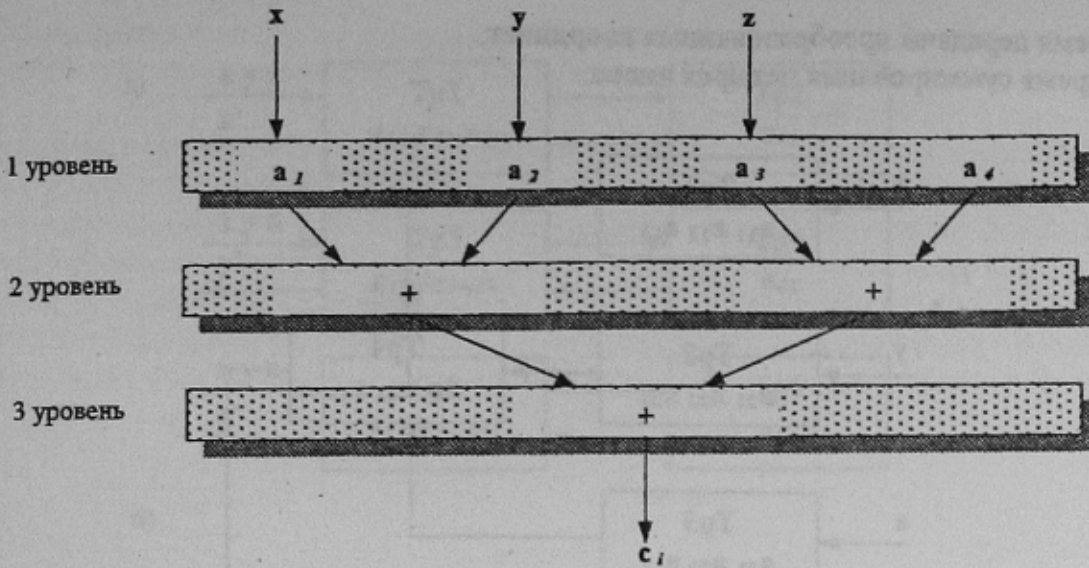


Рисунок 2 – Схема параллельного вычисления координаты

этапа аффинных преобразований, когда в качестве таких процессорных элементов используются транспьютеры [3]. Транспьютер представляет собой 32-разрядный микропроцессор с RISC архитектурой, в состав которого входят внутрикристалльное ЗУПВ, четыре быстродействующих последовательных канала связи (линки), таймер и встроенный процессор обработки с плавающей запятой. Внутренняя архитектура транспьютера, соответствующая фон-неймановским принципам, включает единую мультиплексируемую 32-разрядную шину внешней памяти с диапазоном физических адресов 4 Гбайт.

Взаимодействие каждого транспьютера с другими транспьютерами и периферийными устройствами осуществляется посредством четырех линков, имеющих в составе БИС. Пока ЦПУ занят обработкой, обмен информацией может происходить одновременно по всем четырем последовательным каналам. При времени цикла транспьютера 50 нс на все четыре канала, одновременно работающие со скоростью 10 Мбит/с в обоих направлениях с использованием внутренней памяти, затрачивается примерно 8% производительности ЦПУ.

Рассмотрим структуры параллельного вычисления вектора преобразованных координат c . Применение транспьютера в этих структурах, является наиболее целесообразным, так как в силу своих архитектурных особенностей транспьютер имеет возможность параллельно обрабатывать числа с плавающей запятой и принимать или передавать информацию по линкам.

Первым вариантом распараллеливания является случай, когда вычисление элементов вектора преобразованных координат производится последовательно. Структура приведена на рис. 3.1. Сначала параллельно вычисляются парные произведения, после чего $Tr1$, $Tr2$ и $Tr3$ пересылают свои результаты на $Tr4$, который исполняет роль накапливающего элемента, на котором и производится окончательное вычисление преобразованной координаты. Новые исходные данные для преобразования поступают только после вычисления всего вектора c .

Определим суммарное время преобразования:

$$T1 = t_{пр} + t_{ум} + t_{пер} + 2 * \max(t_{ум}, 3t_{сум}) + 2 * t_{пер} + 3t_{сум} + t_{льцв}$$

где $t_{пр}$ - время передачи исходной координаты;

$t_{ум}$ - время умножения двух чисел;

$t_{пер}$ - время передачи данных между Tr ;

$t_{сум}$ - время суммирования двух чисел;

$t_{\text{выд}}$ - время передачи преобразованных координат;
 $3t_{\text{сум}}$ - время суммирования четырех чисел.

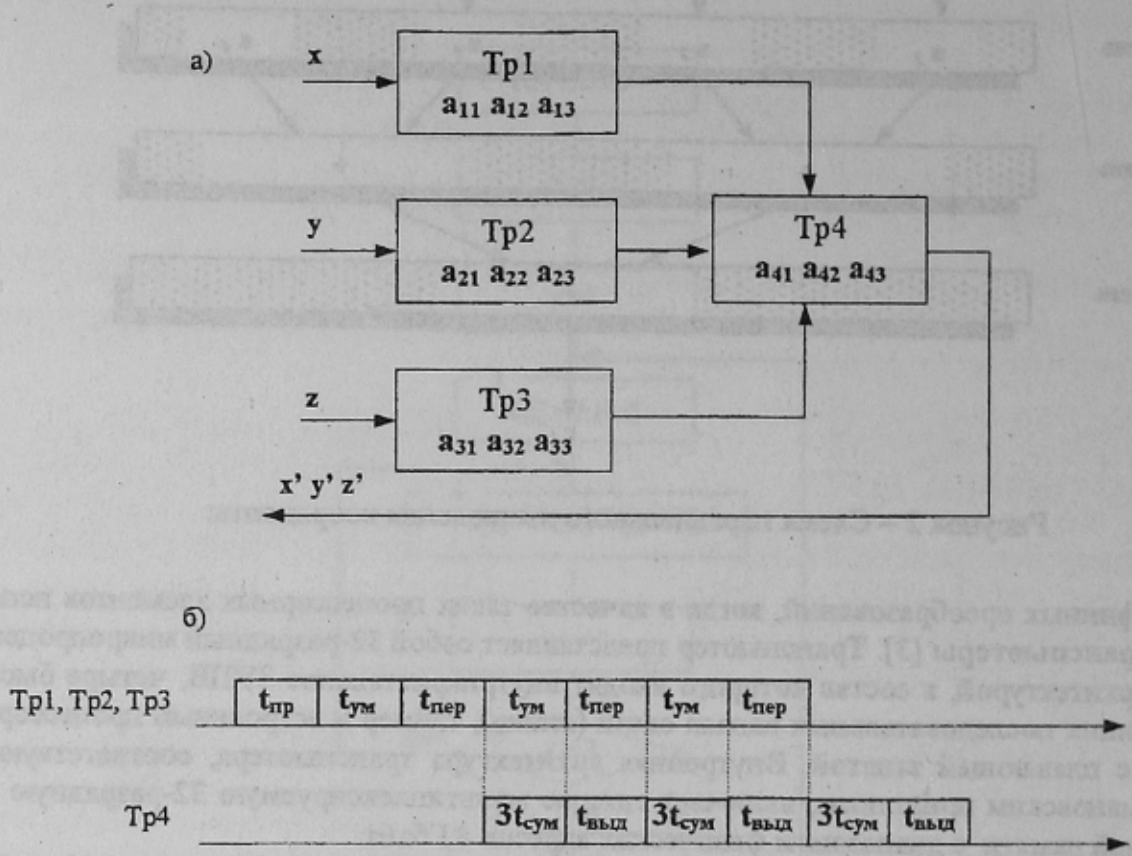


Рисунок 3.1 – а) Структура 1
 б) Временная диаграмма

Времена $t_{\text{пр}}$, $t_{\text{пер}}$ и $t_{\text{выд}}$ являются временами передачи сообщений одинаковой длины между транспьютерами. Поэтому при одинаковой скорости передачи эти времена равны, а в дальнейшем будем обозначать их t . Таким образом,

$$T1 = 5 * t + t_{\text{ум}} + 2 * \max(t_{\text{ум}}, 3t_{\text{сум}}) + 3t_{\text{сум}} \tag{2}$$

Второй вариант параллельной обработки, когда вычисление координат производится параллельно. Структура приведена на рис. 3.2. В приведенной структуре исходный вектор координат поступает на Tr с нечетными номерами последовательно для вычисления каждого парного произведения. Один и тот же элемент вектора передается по *разным* линкам, так как через один линк могут соединяться только 2 транспьютера. Полученное произведение поступает на вход суммирующего элемента (Tr с четными номерами), выполняющего роль накапливающего сумматора, т.е. первое произведение суммируется с a_{4i} , второе – с предыдущей частичной суммой.

После вычисления $a_{3i} * z$, накопленная ранее частичная сумма передается на Tr с нечетными номерами, где и суммируется с третьим парным произведением для получения преобразованной координаты. Конвейер выполняет последовательное вычисление элемента вектора с, но с меньшими временными затратами за счет совмещения операций умножения и суммирования. Эти операции выполняются на разных транспьютерах, поэтому они могут производиться параллельно, но операция умножения в общем случае выполняется дольше.

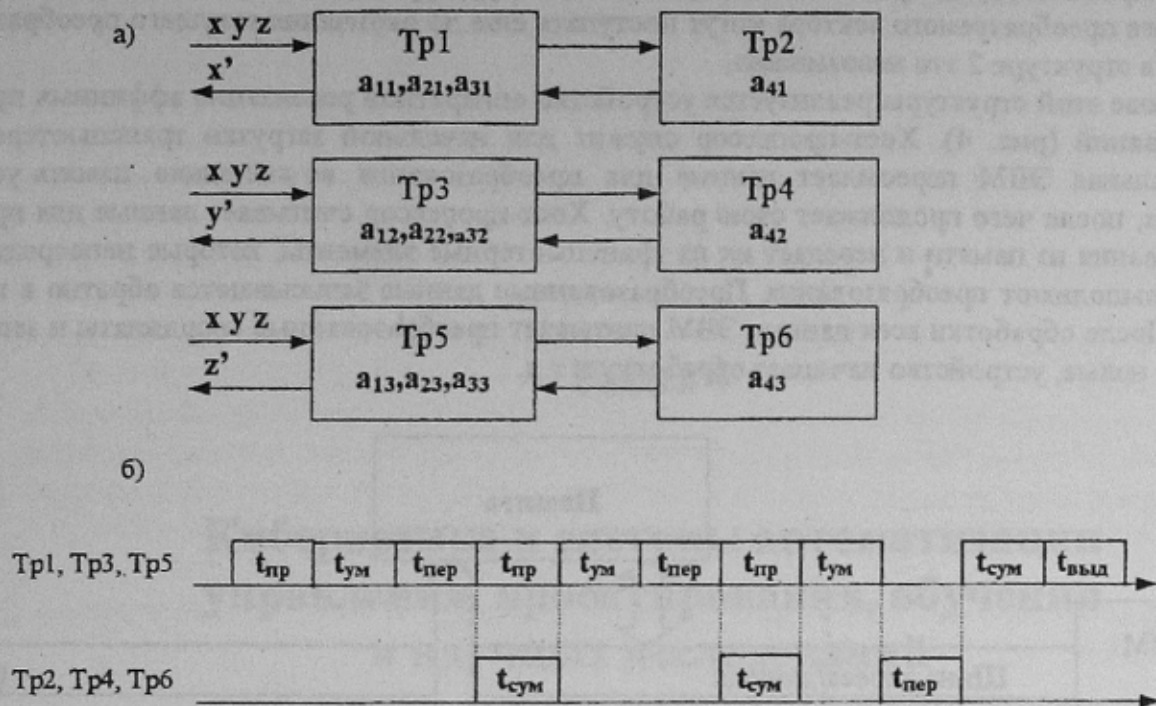


Рисунок 3.2 - а) Структура 2

б) Временная диаграм-

Определим суммарное время преобразования:

$$\begin{aligned}
 T2 &= 3 * t_{пр} + 3 * t_{ум} + 2 * \max(t_{пр}, t_{сум}) + t_{пер} + t_{сум} + t_{выд} = \\
 &= 5 * t + 3 * t_{ум} + 2 * \max(t, t_{сум}) + t_{сум}.
 \end{aligned} \quad (3)$$

Так как два транспьютера выполняют операции параллельно, то в значениях для суммарных времен должно выбираться максимальное время выполнения одной из этих операций.

В общем случае, операция передачи информации между транспьютерами выполняется дольше, а также операция умножения требует большего процессорного времени, чем выполнение трех последовательных суммирований. В связи с этим подкорректируем суммарные времена преобразования для вышеописанных архитектур:

$$\begin{aligned}
 T1 &= 5 * t + 3 * t_{ум} + 3 * t_{сум}, \\
 T2 &= 7 * t + 3 * t_{ум} + t_{сум}.
 \end{aligned}$$

Проанализируем рассмотренные структуры с точки зрения минимальных аппаратных затрат и максимального быстродействия.

Хотя первая структура последовательно выполняет преобразование, но время преобразования $T2$ меньше чем $T1$, так как выше сказано, что $t > t_{сум}$. Также структура 1 содержит на два транспьютера меньше. Таким образом, приблизительная разница составляет:

$$T_{раз} = T2 - T1 = 2 * t - 2 * t_{сум} \quad (4)$$

Исходя из этого, делаем вывод, что в промышленном производстве спецустройства выгоднее использовать вторую структуру.

Кроме того, из временной диаграммы для структуры 1 видно, что элементы следующего преобразуемого вектора могут поступать еще до окончания текущего преобразования, в структуре 2 это невозможно.

На основе этой структуры реализуется устройство аппаратной реализации аффинных преобразований (рис. 4). Хост-процессор служит для начальной загрузки транспьютеров. Центральная ЭВМ пересылает данные для преобразования во внешнюю память устройства, после чего продолжает свою работу. Хост-процессор считывает данные для преобразования из памяти и передает их на транспьютерные элементы, которые непосредственно выполняют преобразования. Преобразованные данные записываются обратно в память. После обработки всех данных ЭВМ считывает преобразованные координаты и записывает новые, устройство начинает обработку и т.д.

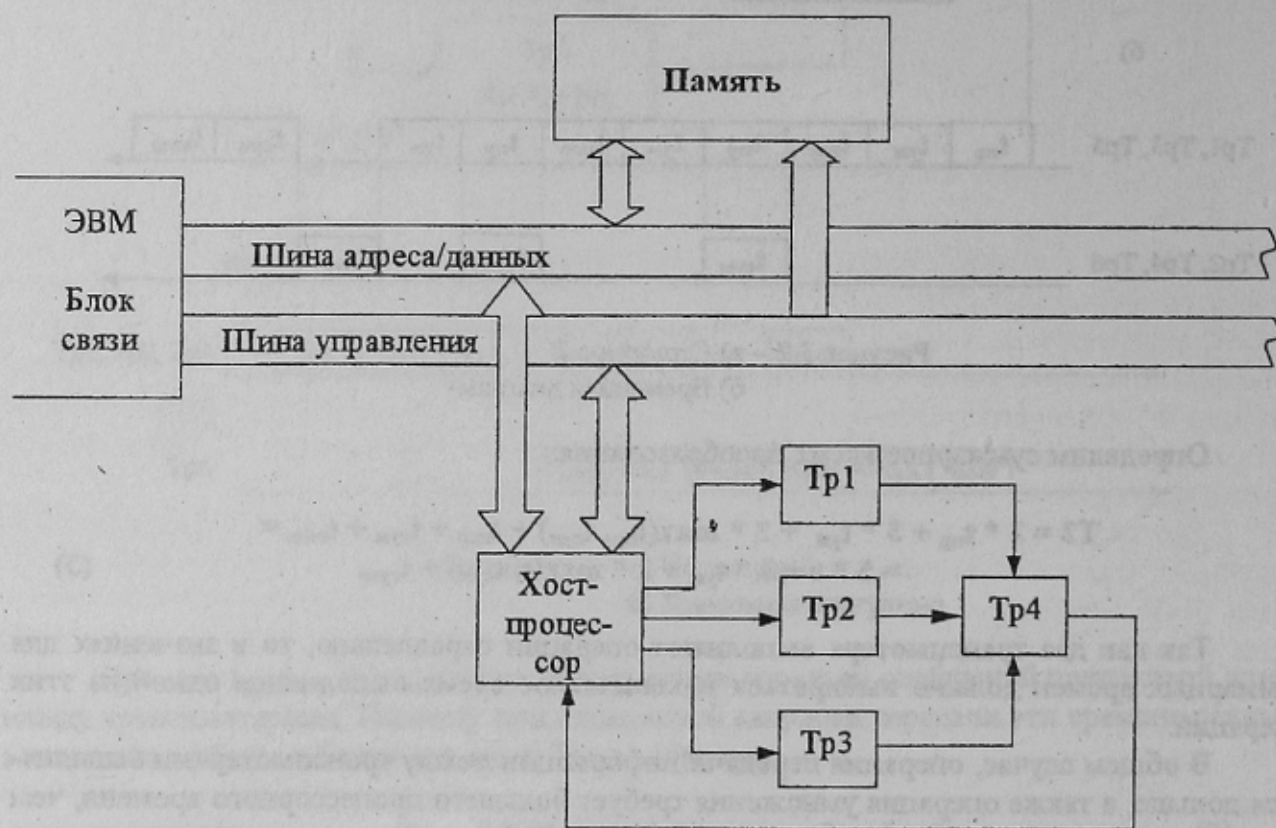


Рисунок 4 – Архитектура спецустройства аппаратной реализации аффинных преобразований

Приведенный анализ показывает, что использование транспьютера с его оригинальной архитектурой в качестве процессорного элемента позволяет реализовать новые параллельные алгоритмы обработки и структуры, что приводит к повышению быстродействия устройств аффинных преобразований.

Литература

1. Ньюмен У., Спрулл Р. Основы интерактивной машинной графики : Пер. с англ. – М.: Мир, 1996.
2. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем : Пер. с англ. – М.: Мир, 1991.
3. Бахтеяров С.Д., Дудников Е.Е., Евсеев М.Ю. Транспьютерная технология – М.: Радио и связь, 1993.