РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я
№ 6 1994

УДК 861.3.06

А. Э. Асмус, А. И. Богомяков, С. И. Вяткин, Ю. М. Попов, Ю. Э. Тиссен, П. И. Унру

(Новосибирск)

ВИДЕОПРОЦЕССОР КОМПЬЮТЕРНОЙ СИСТЕМЫ ВИЗУАЛИЗАЦИИ «АЛЬБАТРОС»

Описывается архитектура и принципы работы видеопроцессора компьютерной системы визуализации «Альбатрос». Отмечается роль предложенного авторами эффективного метода растрирования многоугольников для создания однородной конвейерной архитектуры. Показано, что в рамках этой архитектуры решаются проблемы удаления невидимых поверхностей, фильтрации изображения и вычисления цвета. Анализируется влияние механизма многоуровневого маскирования областей экрана на производительность видеопроцессора.

Введение. Основными показателями, определяющими реализм и качество изображений в системах визуализации, традиционно являются количество аппроксимирующих многоугольников, обрабатываемых системой за время телевизионного кадра, и разрешение экрана, задаваемое полным количеством пикселов экрана. Кроме того, реализм изображений определяется наличием в системе генерации специализированных эффектов: текстуры, затенения, тумана и др., а качество изображений зависит от способа и параметров фильтрации, т. е. процесса устранения эффектов дискретности телевизионного растра. Из двух основных узлов системы визуализации: геометрического и видеопроцессоров — наибольший объем вычислений, отвечающих за приведенные выше показатели, приходится на видеопроцессор. Действительно, видеопроцессор осуществляет растрирование многоугольников, удаление невидимых поверхностей, вычисление цвета в каждом видимом пикселе. В первом приближении можно сказать, что количество вычислений в видеопроцессоре пропорционально количеству пикселов экрана. При этом только для вычисления цвета на каждый пиксел требуются десятки арифметических операций [1]. Проблемы, возникающие при разработке видеопроцессора, заключаются, с одной стороны, в создании эффективных алгоритмов, а с другой — в способах распараллеливания вычислений. Полученная в результате архитектура должна одновременно удовлетворять требованиям к объему оборудования, производительности и качеству изображений.

Важным свойством архитектуры, облегчающим создание, тестирование и ремонт видеопроцессора, является ее однородность, означающая, что в состав видеопроцессора многократно входят однотипные элементарные модули. Вместе с тем часто возникает необходимость создавать различные варианты видеопроцессора, включающие разное количество элементарных модулей, под определенные требования производительности и качества изображения.

Ранее были предложены различные варианты архитектур видеопроцессора, различающиеся способом растрирования многоугольников, удаления невидимых поверхностей, вычисления цвета (см., например, [3, 4]). Видеопроцессор системы «Альбатрос» [2] отличается рядом преимуществ, которые позволяют рассматривать его архитектуру как перспективную, в частности, для воплощения ее на новой элементной базе с использованием технологии заказных СБИС.

Принципы построения. В основу архитектуры видеопроцессора системы «Альбатрос» положена идея рекурсивной процедуры деления экрана, локализующей внутреннюю область многоугольника в процессе растрирования. В обобщенном виде метод растрирования многоугольников изложен в [5]. Основными характерными чертами метода являются описание многоугольников наборами прямых, проходящих через ребра, и рекурсивное деление экрана на клетки, площадь которых уменьшается в 4 раза с каждым шагом деления.

В системе «Альбатрос» с целью удобства аппаратной реализации на доступной элементной базе представление ребер уравнениями прямых общего вида было заменено на специальное описание, включающее два параметра и два признака. Такое задание позволяет определить положение всех четырех клеток при делении относительно ребра, используя одну арифметическую операцию — сложение. Алгоритм реализован как конвейер из идентичных элементарных вычислительных каскадов (процессорных элементов), где каждый каскад осуществляет шаг рекурсии для клеток определенного размера («уровня»).

Приоритетная последовательность граней при обработке и маскирование участков экрана обеспечивают удаление невидимых поверхностей. Многоугольники и их части, попавшие в замаскированные участки экрана, исключаются из обработки на ранних стадиях деления экрана.

С целью достижения необходимой производительности видеопроцессора деление экрана в конвейере продолжается не до пиксела, а до клетки размером 4×4 пиксела. Затем в следующем вычислительном каскаде конвейера определяется положение фрагмента многоугольника в клетке одновременно относительно всех 16 пикселов. Для устранения дефектов, связанных с дискретностью телевизионного растра, определение принадлежности элемента изображения многоугольнику осуществляется на повышенном разрешении растра: каждый пиксел разбит на 16 субпикселов. Определение факта накрытия фрагментом многоугольника субпикселов клетки осуществляется таблично. Это возможно благодаря тому, что разрядность параметров ребер в клетке 4×4 может быть снижена по сравнению с входной разрядностью параметров. В результате формируется субпиксельный код фрагмента.

Полученный код фрагмента параллельно сравнивается с кодом маски клетки, сформированным из ранее обработанных более приоритетных многоугольников. В коде фрагмента остаются отмеченными только видимые субпикселы, а код маски дополняется от вновь замаскированных субпикселов. Память масок модифицируется, а субпиксельный код фрагмента сворачивается в площади пикселов и передается в четыре идентичных канала вычисления цвета. Цвет вычисляется исходя из типа многоугольника (интерполированный, текстурированный) и присутствия дымки (тумана).

Координаты, цвет и площадь видимой части многоугольника в пикселе поступают в блок видеобуфера, который содержит память для цвета всех пикселов экрана (по одному банку на каждый канал вычисления цвета) и снабжен схемой накопления цвета. Результирующий цвет пиксела есть взвешенная сумма цветов видимых фрагментов многоугольников, попавших в пиксел. Накопленное в видеобуфере изображение считывается синхронно со строчной разверткой монитора. Цветовые коды проходят через схему «грубого» фильтра и гамма-коррекции и поступают на цифроаналоговые преобразователи (DAC), вырабатывающие сигналы управления лучами ЭЛТ. В данной работе дано описание базового варианта видеопроцессора, включающего блоки растрирования, вычисления цвета и видеобуфера.

Блок растрирования. Линейный приоритетно упорядоченный список многоугольников и огней с атрибутами, заданными в виде линейных функций, поступает из геометрического процессора (ГП) (см. [6]) и запоминается во входном двойном буфере (рис. 1), называемом памятью кадра (FB). В то время как в одной половине FB происходит прием данных, из другой данные считываются и передаются в конвейер видеопреобразования. В случае превышения размера массива данных над размером банка FB переходит в режим памяти

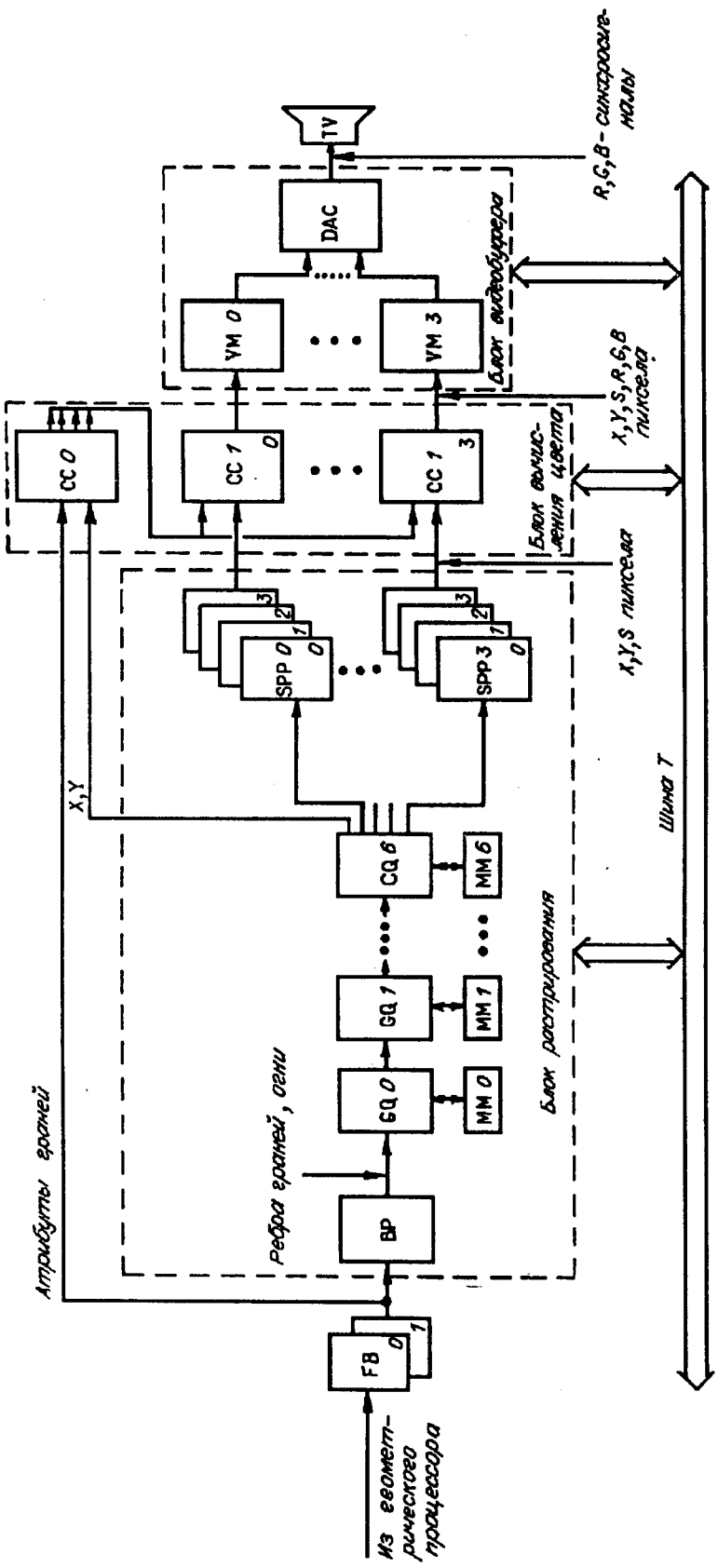
Из геометрического процессора
FB
BP
Ребра граней, огни
Атрибуты вершин
X,Y
SPP 0
SPP 3
CC 0
CC 1
VM 0
VM 3
DAC
TV
GQ 0
GQ 1
GQ 6
MM 0
MM 1
MM 6
Блок растрирования
Шина T
X,Y,S пиксела
Блок вычисления цвета
X,Y,S,R,G,B пиксела
Блок видеоадаптера
R,G,B - синхросигналы

Рис. 1

«магазинного» типа (FIFO) и тем самым не вносит принципиального ограничения на размер списка обрабатываемого кадра.

Чтение данных из FB осуществляется с помощью модуля предобработчика (BP). Параметры ребер многоугольников передаются в конвейер процессоров растрирования, а атрибуты многоугольников — в блок вычисления цвета. Геометрический примитив типа «огонь» в модуле BP аппроксимируется ребрами, так что в дальнейшем их обработка ничем не отличается от обработки многоугольников.

Как отмечалось, в качестве элемента разложения многоугольников в конвейере растрирования используется рекурсивно уменьшающаяся клетка, первоначальный размер которой равен площади экрана, а конечный — 4 × 4 пиксела. Многоугольник на всех шагах рекурсивного деления описывается набором линий, проходящих через ребра. Описание ребра включает два признака и два числа. По одному из признаков, обозначенному TG, ребра классифицируются на вертикальные и горизонтальные по углу наклона к оси X: для вертикальных — угол больше 45°, для горизонтальных — меньше или равен 45°. Другой признак (LR) показывает, с какой стороны от ребра лежит выпуклый многоугольник, т. е. правым или левым является ребро многоугольника. Два параметра задают координаты пересечения линии с краями экрана (клетки) или их продолжениями (рис. 2): для вертикальных ребер (X_1, X_2) — X-координаты пересечения с линиями, продолжающими верхнюю и нижнюю границу экрана (клетки) ($Y = \pm 1$), для горизонтальных ребер (Y_1, Y_2) — Y-координаты пересечения с линиями, продолжающими левую и правую границу экрана (клетки) ($X = \pm 1$).

Такое задание позволяет определить положение всех четырех клеток при делении относительно ребра с использованием одной арифметической операции — сложения. Действительно, полусумма координат пересечения задает положение средней точки на отрезке линии между противоположными границами (или их продолжениями) экрана (клетки), а ее старшие разряды вместе со старшими разрядами координат пересечения и парой признаков достаточны для определения положения клеток относительно линии ребра при делении экрана.

Рекурсивная процедура деления на клетки осуществляется в ряде идентичных процессорных элементов, называемых клеточными процессорами (GQ) и включенных в конвейер. В каждом GQ осуществляется деление клетки данной площади на четыре равные клетки в 4 раза меньшей площади. Это означает, что, во-первых, определяется принадлежность фрагмента многоугольника в клетке клеткам меньшего размера и, во-вторых, формируется описание фрагментов многоугольника в клетках меньшего размера. У каждого GQ имеется так называемая память масок (MM), где хранится код маски, в котором помечаются соответствующим битом целиком занятые клетки экрана данного размера.

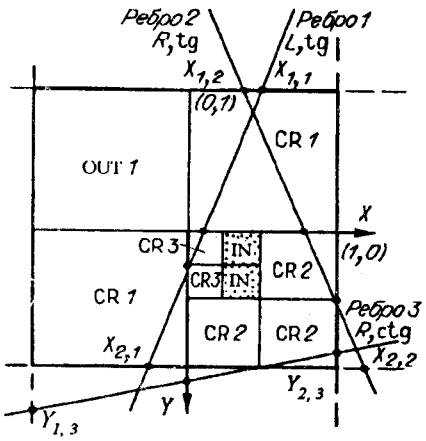

Рис. 2

Для определения расположения каждой из четырех клеток относительно фрагмента многоугольника для каждого ребра формируются 4-битные коды:

CI_i — код «внутренних» клеток ребра i: бит установлен в единицу, если линия, проходящая через ребро, не пересекает соответствующую клетку и клетка лежит с той же стороны от линии, что и весь многоугольник;

CRI_i — код пересеченных клеток ребра i плюс «внутренних».

Коды CI_i, CRI_i определяются табличным методом с использованием ПЗУ. На вход таблицы поступают старшие

разряды двоичного представления величин:

$$V_1,\ V_2,\ (V_1 + V_2)/2, \tag{1}$$

а также признаки TG и LR. В выражении (1) под символом V понимается X или Y в зависимости от признака TG.

Для получения кода внутренних клеток фрагмента многоугольника CI 4-битовые коды CI_i ребер объединяются по схеме И. Аналогично для получения кода пересеченных и внутренних клеток фрагмента CRI коды CRI_i также объединяются по схеме И. Объединение происходит по мере поступления ребер до тех пор, пока не поступит последнее ребро фрагмента многоугольника в клетке. Затем полученные коды объединяются с кодом маски M для исключения из рассмотрения полностью замаскированных клеток:

$$CRI = \overline{M} \,\&\, \overset{n}{\underset{i=1}{}} CRI_i, \tag{2а}$$

$$CI = \overline{M} \,\&\, \overset{n}{\underset{i=1}{}} CI_i, \tag{2б}$$

$$CR = CRI * \overline{CI}, \tag{2в}$$

$$M_n = M + CI. \tag{2г}$$

В логических выражениях (2) символы "*", "+" и "—" означают операции И, ИЛИ и инверсию; M — код, считываемый из памяти масок; CR — код только пересеченных клеток; M_n — обновленный код маски, который заносится в MM по окончании обработки фрагмента вместо M.

Таким образом, по окончании обработки ребер многоугольника текущей клетки возможны следующие варианты (см. рис. 2):

1. Многоугольник не принадлежит клетке (клетка OUT). В этом случае данная область исключается из рассмотрения при дальнейшей растризации этого многоугольника.

2. Многоугольник полностью покрывает клетку (клетка IN). Клетка IN подвергается дальнейшему разбиению, а область экрана объявляется занятой, чтобы фрагменты следующих многоугольников в данной области не участвовали в построении изображения.

3. Многоугольник частично перекрывает клетку (клетка CR). В этом случае клетка с ребрами, принадлежащими ей, подвергается дальнейшему разбиению.

Процесс деления областей экрана можно представить как обход кватернарного дерева (рис. 3). Очевидно, что наибольший объем вычислений приходится на нижние уровни деления экрана, на которых необходимо распараллеливание. Исходя из возможности аппаратной реализации и требуемой производительности был реализован следующий вариант: с нулевого по шестой уровни деления ветви дерева обходятся последовательно, а на последнем уровне параллельно обрабатывается сразу 16 клеток, размер которых равен пикселу.

Использование многоуровневой памяти масок позволяет эффективно отбраковывать многоугольники или их фрагменты, попавшие в уже занятые более близкими многоугольниками области экрана. Тем самым уменьшается время обработки сцены. Отметим также, что зависимость времени обработки от глубинной сложности отображаемой сцены становится при многоуровневом маскировании нелинейной: увеличение глубинной сложности не приводит к существенному увеличению времени обработки сцены. Производительность видеопроцессора в большей степени определяется суммарным периметром многоугольников в сцене. Статистические исследования показали, что для большинства графических баз данных производительность системы «Альбатрос» определяется геометрическим процессором, в то время как видеопроцес-

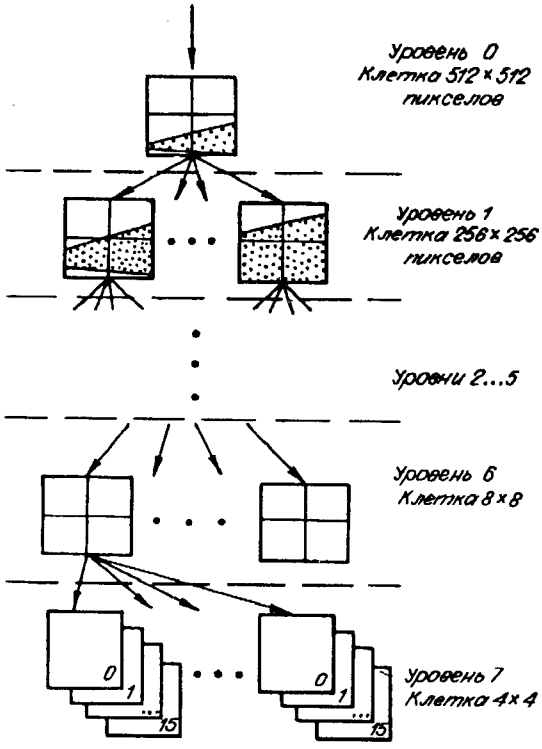

Рис. 3

сор потенциально имеет гораздо большую скорость обработки и не вносит задержки в систему. На рис. 4 показаны графики зависимости времени обработки нескольких сцен от наличия ММ на разных уровнях деления. Из графиков видно, что все сцены при наличии только ММ последнего уровня не могут обработаться за время полукадра (20 мс). При включении верхних уровней ММ время обработки существенно уменьшается. Например, сцена ROMBM содержит всего 22 многоугольника, уложенных в стопку, и требует более 25 мс для ее обработки без многоуровневого маскирования, что существенно больше, чем для сцен с гораздо большим числом геометрических примитивов, но с меньшей глубинной сложностью. При включении верхних уровней маскирования время обработки сцены ROMBM составило чуть более 12 мс. На рисунке *I* — маскирование только на 7-м уровне, *II* — на 6, 7-м уровнях, *III* — на 5—7-м уровнях, *IV* — на 1—7-м уровнях.

Для эффективного поиска занятых пикселов на периметре многоугольников фрагмент многоугольника в клетке размером 4 × 4 пиксела из последнего в конвейере GQ передается в 16 параллельно работающих процессоров, называемых субпиксельными процессорами (SPP). Каждый SPP оперирует одним пикселом, разбитым на 16 субпикселов. В SPP выполняется заключительная операция поиска занятых субпикселов и их маскирования в субпиксельной ММ. С этой целью координаты V_1, V_2 ребер фрагмента многоугольника в клетке, взятые с разрядностью, обеспечивающей точность в половину размера субпиксела, вместе с признаками TG и LR подаются на вход табличной памяти. Параллельный код на выходе таблицы определяет выборку из субпикселов, лежащих с той же стороны ребра, где расположен весь многоугольник. Для уменьшения объема памяти таблицы используются свойства симметрии субпиксельной выборки и предварительное перекодирование входной информации. Субпиксельные коды последовательно поступающих ребер объединяются по схеме И для

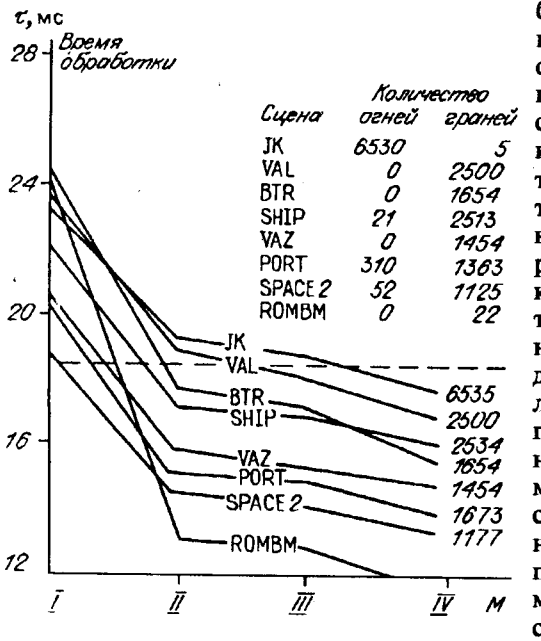

Рис. 4

получения субпиксельного кода фрагмента многоугольника. Отметим, что субпиксельные выборки определяются таким образом, чтобы суммарная площадь занятых (отмеченных) субпикселов была максимально приближена к геометрической площади фрагмента многоугольника. Значение площади в дальнейшем используется при суммировании цветового вклада всех фрагментов многоугольников, попавших в пиксел.

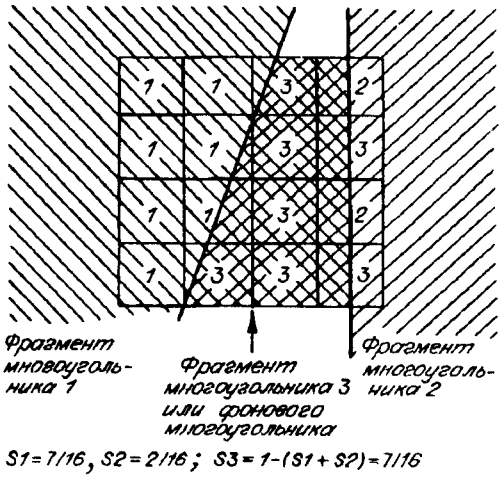

Рис. 5

Субпиксельный код фрагмента многоугольника в пикселах клетки параллельно сравнивается с кодом субпиксельной маски, считываемой из памяти масок нижнего уровня. Количество ячеек этой памяти равно количеству клеток 4×4 на экране, а полная длина слова — $16 \times 16 = 256$ бит. Адресом в память является двоичное разрядное представление X- и Y-координат клетки на экране. Результат сравнения есть код, представляющий выборку видимых субпикселов, а также обновленный код маски для модификации памяти масок.

Далее субпиксельный код видимого фрагмента многоугольника сворачивается в каждом субпиксельном процессоре в площадь занятой многоугольником части пиксела. Площадь вычисляется простым определением количества занятых субпикселов в пикселе (рис. 5).

Затем видимые пикселы, полученные на 16 субпиксельных процессорах, распределяются по четырем каналам вычисления цвета (СС). В каждый канал поступают видимые пикселы (если они имеются) из четырех субпиксельных процессоров, соответствующих такой выборке в клетке, которая приводит в среднем к максимально равномерному по каналам потоку данных. В СС также поступают из ВР коэффициенты линейных уравнений, по которым вычисляются значения интенсивности (I) и дальности (Z). Процесс вычислений в СС также имеет конвейерную организацию. В модифицированном варианте системы «Альбатрос-Т» блок СС дополнен оборудованием, которое обеспечивает обработку текстурированных поверхностей. Подробное описание организации процесса вычислений в блоке СС приведено в [7].

Блок видеобуфера. Данные о цвете пиксела с длиной слова 24 разряда (по 8 бит на компоненты R, G, B), его адрес, а также площадь фрагмента передаются из СС по четырем параллельным каналам в блок видеобуфера (рис. 6). Блок видеобуфера включает в себя четыре параллельных модуля, содержащих память изображения (VM_i), а также модули цифроаналогового преобразования. Модули VM состоят из устройства фильтрации F1 и сдвоенной 24-разрядной видеопамяти VMEM. В устройстве фильтрации F1 вычисляется произведение цвета фрагмента многоугольника на площадь, которую он занимает в пикселе, и происходит накопление результата в VMEM. Следует отметить, что с целью уменьшения разрядности VMEM и сохранения динамического диапазона в памяти хранится логарифм яркости каждого цвета. Этим учитывается нелинейный характер восприятия глазом человека пороговых градаций яркости: наибольшую чувствительность глаз имеет в диапазоне малой яркости, которая уменьшается с ростом яркости. В связи с этим все вычисления в устройствах F1, F2 производятся с учетом этой нелинейной шкалы. Так, схема вычисления суммы вкладов фрагментов многоугольников в пикселе имеет вид:

1) умножение на площадь фрагмента («взвешивание») в логарифмической шкале заменяется на сложение:

$$\overline{C}' = \overline{C} + (1/\alpha)\ln S, \qquad (3)$$

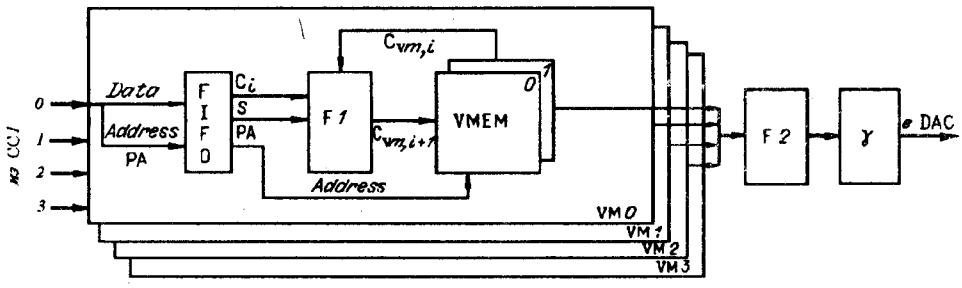

Рис. 6

2) суммирование вкладов при накоплении заменяется на сложение, вычитание и табличную функцию:

$$\overline{C}_{vm} = \overline{C}_2' + (1/\alpha)\ln\{1 + \exp(\alpha d\overline{C})\}, \tag{4а}$$

$$d\overline{C} = \overline{C}_1' - \overline{C}_2'. \tag{4б}$$

В выражениях (3), (4) $\overline{C}$ — трехкомпонентный *RGB*-вектор, обозначающий цвет многоугольника в пикселе; C' — вклад в результирующий цвет пиксела; C_{vm} — результирующий цвет пиксела от вкладов C_1, C_2; S — площадь фрагмента; α — коэффициент, который характеризует степень нелинейности шкалы и выбирается с учетом характеристики отображающего устройства.

При заполнении видеопамяти количество обращений по данному адресу равно количеству видимых фрагментов в соответствующем этому адресу пикселе. Каждая ячейка видеопамяти выполняет также функцию накопительного регистра. Поэтому перед началом обработки сцены необходима чистка памяти. Однако более предпочтительным является формирование признака первой записи в данную ячейку. Признак формируется в субпиксельном процессоре с использованием памяти масок. Цвет фрагмента с этим признаком записывается без изменения. Последующие фрагменты того же пиксела проходят цикл чтение—модификация—запись.

В то время как в одной части VMEM накапливается изображение, из другой происходит чтение данных синхронно со строчной разверткой и передача их в устройство «грубой» фильтрации F2. Устройство присваивает каждому пикселу новый цвет, который есть среднее арифметическое четырех цветов: цвета пиксела и цветов трех его соседей — правого, нижнего и касающегося правого нижнего угла пиксела. Действие такого фильтра на изображение эквивалентно действию прямоугольного фильтра ("box"-фильтр) с апертурой 2 × 2 пиксела. Наблюдения показали, что фильтр устраняет эффект чересстрочного мелькания и уменьшает элайзинг при незначительном падении разрешения. Вычисления в устройстве F2 проводятся в логарифмической шкале в два этапа: сначала вычисляются полусуммы яркостей в соседних пикселах из соседних строк (устройство имеет поэтому буфер на строку), затем полусуммы соседних пикселов на строке.

Для того чтобы скомпенсировать нелинейную зависимость яркости пятна, высвечиваемого лучом ЭЛТ, от управляющего напряжения, значения компонент цвета поступают из устройства фильтрации F2 на таблицу гамма-коррекции. Таблица гамма-коррекции содержит значения композиции двух функций: антилогарифма, переводящего логарифмическую шкалу яркостей в линейную, и собственно компенсирующей функции, которая может быть определена экспериментально. Контроль неисправностей каждого модуля видеопроцессора осуществляется от управляющей ЭВМ через шину Т. Для этой цели в каждый модуль встроено дополнительное тестовое оборудование.

Заключение. Алгоритмические и аппаратные решения, реализованные в видеопроцессоре системы «Альбатрос», позволили создать надежную высокопроизводительную систему синтеза визуальной среды. Наиболее важными из них являются:

— предложенный авторами эффективный механизм растрирования многоугольников, позволяющий с помощью простой рекурсивной процедуры находить пикселы экрана, принадлежащие многоугольнику;

— механизм многоуровневого маскирования многоугольников, основанный на том, что многоугольники (или их части), попавшие в участки экрана, накрытые ранее более близкими к наблюдателю многоугольниками, могут не обрабатываться. Такой механизм увеличивает производительность системы и позволяет эффективно обрабатывать сцены с большой глубинной сложностью;

— высокая степень однородности архитектуры: видеопроцессор состоит из небольшого числа типов элементарных процессоров, каждый из которых может многократно входить в полную архитектуру системы. Так, базовый вариант видеопроцессора системы включает семь клеточных процессоров, 16 субпиксельных процессоров, четыре вычислителя цвета и четыре модуля видеопамяти. Однородность архитектуры упрощает изготовление, настройку и тестирование системы, повышает показатели надежности, позволяет строить различные варианты системы, удовлетворяющие различным требованиям производительности и качества;

— сочетание двух видов фильтрации изображения: «тонкой» на субпиксельном уровне и «грубой» с апертурой 2 × 2 пиксела. Введение апертуры пиксела при фильтрации на субпиксельном уровне привело бы к существенному усложнению клеточного и субпиксельного процессоров. Кроме того, снизилась бы производительность системы, так как при расширенной апертуре увеличилось бы количество фрагментов многоугольников в каждом пикселе. «Грубый» фильтр 2 × 2 пиксела дает эффект, приближенный к идеальному фильтру с расширенной апертурой, с минимальными затратами;

— использование логарифмической шкалы яркостей, позволяющее экономно кодировать уровни яркости основных цветов в соответствии с нелинейным характером чувствительности глаза, что приводит к снижению разрядности вычислений цвета и видеопамяти.

СПИСОК ЛИТЕРАТУРЫ

1. **Ковалев А. М., Талыкин Э. А.** Машинный синтез визуальной обстановки // Автометрия.—1984.—№ 4.
2. **Долговесов Б. С.** Семейство компьютерных систем визуализации «Альбатрос» // Автометрия.—1994.—№ 6.
3. **Башков Е. А., Казак А. В.** Генераторы изображения для авиатренажеров // Зарубеж. радиоэлектрон.—1984.—№ 8.
4. **Богданов В. В., Ковалев А. М., Нефедов И. Б. и др.** Канал видеопреобразования синтезирующей системы визуализации // Автометрия.—1986.—№ 4.
5. **Вяткин С. И., Долговесов Б. С., Мазурок Б. С., Рожков А. Ф.** Эффективный метод растрирования изображений для компьютерных систем визуализации реального времени // Автометрия.—1993.—№ 5.
6. **Долговесов Б. С., Мазурок Б. С., Маслобоев Ю. В., Рожков А. Ф.** Геометрические преобразования в семействе «Альбатрос» // Автометрия.—1994.—№ 6.
7. **Мазурок Б. С., Рожков А. Ф., Сальников Ю. А. и др.** Генерация текстурированных поверхностей и специализированных эффектов в системах «Альбатрос» // Там же.

Поступила в редакцию 23 августа 1994 г.