

ОБЪЕДИНЕННЫЙ  
ИНСТИТУТ  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ  
ДУБНА

2019/83

18/4-83

P10-83-7

Х.Лайх, Ф.В.Левчановский, В.И.Приходько

МУЛЬТИМИКРОПРОЦЕССОРНАЯ СИСТЕМА  
УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫМ  
ГРАФИЧЕСКИМ ТЕРМИНАЛОМ  
Архитектура системы

Направлено в журнал  
"Техника связи и электроника"

1983

## 1. ВВЕДЕНИЕ

За последние годы значительно увеличилось количество разработок и расширился выпуск современных диалоговых средств. В настоящее время выпускается широкий спектр диалоговых устройств, начиная от простых для отображения алфавитно-цифровой информации и кончая мощными интеллектуальными графическими терминалами, которые используются в системах обработки данных, в системах разработки на базе ЭВМ /CAD – computer aided design/ и в системах управления производством /CAM – computer aided manufacturing/, обеспечивая при этом значительный эффект как в сфере научных исследований и в производстве, так и в других областях человеческой деятельности.

Ожидается, что за период 1978-84 гг. выпуск дисплейных устройств будет увеличиваться с каждым годом на 35%/1/, в частности, предполагается, что в США число используемых терминалов возрастет с 69000 в 1978 году до 454000 в 1984 году.

В ОИЯИ дисплейные устройства используются уже с середины 60-х годов. За последние годы в ОИЯИ создан ряд совместимых устройств для построения дешевых и компактных графических систем/2/:

- быстродействующий регенеративный дисплей со скоростью вывода информации до  $10^6$  точек/с, предназначенный главным образом для использования в спектрометрии и в траекторных измерениях;
- дисплей на запоминающей ЭЛТ 31ЛН3;
- универсальный дисплейный процессор в стандарте КАМАК для управления указанными устройствами;
- световой карандаш и координатный шар для оперативного взаимодействия и ввода в ЭВМ графической информации с экрана дисплея;
- асинхронный и синхронный модемы для подключения терминалов к центральной ЭВМ или к локальной терминальной сети по телефонным линиям связи;
- графический терминал на запоминающей ЭЛТ 31ЛН4 со встроенной микро-ЭВМ/3/.

На основе опыта этих разработок и тенденций развития графических систем в настоящее время разработан и создан в модульном исполнении прототип интеллектуального графического терминала /ИГТ/, который имеет высокую производительность и дополняет существующее семейство устройств.

## 2. ТЕРМИНОЛОГИЯ

В соответствии с /4,5/ в дальнейшем изложении используются следующие понятия:

Дисплей: устройство или прибор для визуального представления алфавитно-цифровых /буквы, знаки, символы/ и графических /линии, точки, кривые, .../ данных.

Графические примитивы: элементы, из которых состоит изображенная на дисплее картина /точка, линия, символ, .../.

Графический объект: адресуемый элемент картины, объединяющий несколько примитивов /называется также сегментом/.

Мировые координаты /МК/: независимые от конкретного устройства координаты, с помощью которых пользователь /программа пользователя/ описывает графические примитивы и преобразования.

Дисплейные координаты /ДК/: координаты конкретной дисплейной системы /например, адресуемые точки экрана дисплея/.

Дисплейный файл: описание картины на основе ДК.

Поле изображения: заданное окно в системе ДК.

Окно: заданная часть в поле МК.

Преобразование визуализации: преобразование, которое переводит заданную часть в поле МК /окно/ в заданное поле изображения.

Графическое преобразование: перемещение, масштабирование и поворот графических объектов.

## 3. АРХИТЕКТУРА МУЛЬТИМИКРОПРОЦЕССОРНОЙ СИСТЕМЫ /ММПС/ УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫМ ГРАФИЧЕСКИМ ТЕРМИНАЛОМ

### 3.1. Функции специализированной ММПС. Распределение функций

При определении спектра функций ММПС были заданы следующие условия:

- повышение производительности по отношению к уже существующим устройствам;
- совместимость с существующими устройствами;



- тенденция к стандартизации графических систем<sup>/4/</sup>;
- использование самой современной микроэлектроники, что позволяет значительно увеличить внутренние вычислительные мощности терминала и перенести выполнение ряда функций с центральной ЭВМ на терминал.

Исходя из этого для ММПС был определен следующий набор функций:

1. Работа с центральной ЭВМ через последовательный и/или параллельный интерфейсы.
2. Управление диалогом пользователь-терминал и ЭВМ-терминал.
3. Предоставление возможности манипуляции графическими объектами /создание, модификация и стирание объектов в файле/.
4. Ввод графических данных.
5. Построение графических примитивов на экране векторного дисплея по командам, содержащимся в дисплейном файле.
6. Выполнение специальных графических функций /графические преобразования, отсечение невидимых линий, обработка графических объектов и их атрибутов/.
7. Хранение картин пользователя в МК, а также той части дисплейного файла, которая выводится в режиме регенерации /регенеративная часть/.
8. Инициализация всей ММПС после включения, контроль за работой системы и составление статусной информации о ее работе.

Указанные функции выполняются ММПС, состоящей из процессора-монитора, модуля общей памяти, дисплейного модуля и арифметического модуля /рис.1/. При этом распределение задач производится следующим образом/6/:

- процессор - монитор - задачи 1,2,3 и 8;
- модуль общей памяти - задача 7;
- дисплейный модуль - задачи 4,5;
- арифметический модуль - задача 6.

Одним из важных условий, принятых при разработке ММПС, является возможность расширения системы путем включения в нее дополнительных модулей.

### 3.2. Структура ММПС

#### 3.2.1. Общая шина

<sup>/7/</sup> показано, что связь между модулями ММПС при помощи общей шины /ОШ/ обеспечивает по сравнению с другими способами наилучшее отношение стоимости связи к производительности системы. Простая техническая реализация этой структуры способствовала широкому распространению ОШ.

Однако если к общей шине подключить более 4 активных модулей, то производительность системы может уменьшаться из-за конфликтов

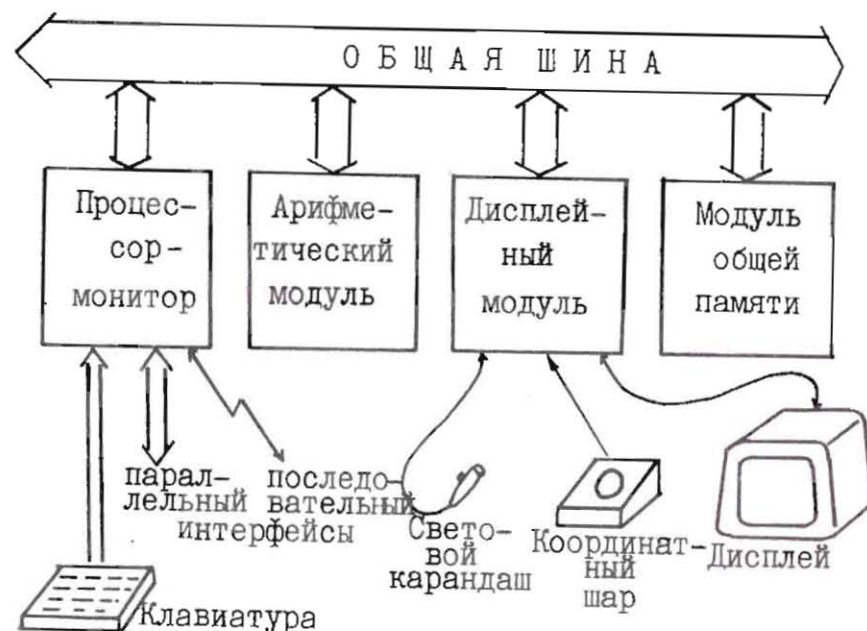


Рис.1. Структура мультимикропроцессорной системы.



Рис.2. Структура арбитра общей шины.

при доступе к ОШ. Поэтому при разработке системы необходимо:

1/ минимизировать число обращений к ОШ со стороны любого модуля, то есть модули должны обрабатывать присвоенные им задачи как можно более автономно;

2/ свести к минимуму время захвата ОШ для выполнения элементарной операции /чтения или записи/;

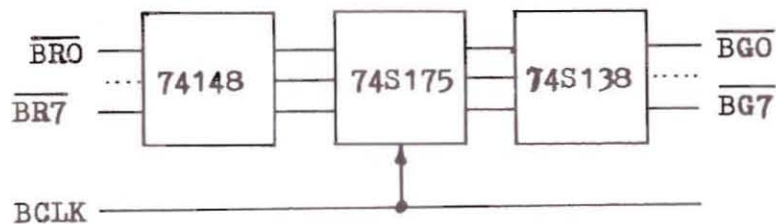


Рис.3. Схемная реализация арбитра общей шины.

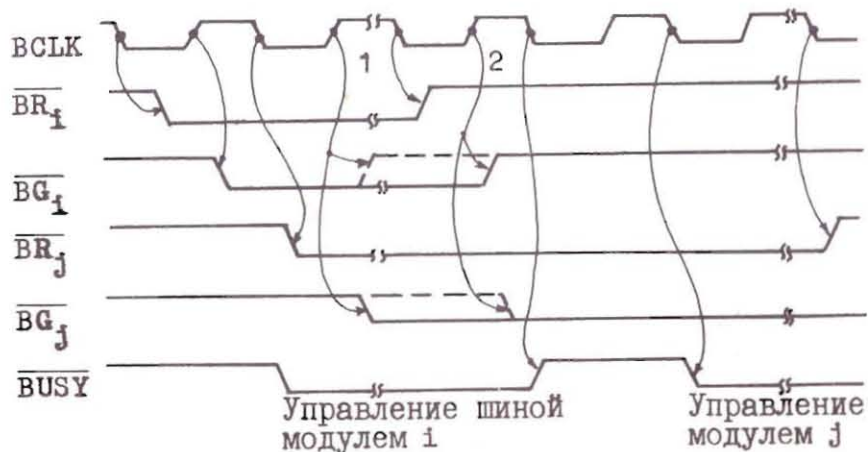


Рис.4. Временная диаграмма работы арбитра: 1/  $BR_j$  имеет более высокий приоритет, чем  $BR_i$ ; 2/  $BR_i$  имеет более высокий приоритет, чем  $BR_j$ .

3/ использовать арбитр для разрешения конфликтов доступа при обращении к ОШ, при этом работа арбитра не должна существенно увеличивать время выполнения элементарных операций.

На рис.2 показана структура используемого в рассматриваемой ММПС арбитра общей шины, а на рис.3 - его техническая реализация. Временная диаграмма работы арбитра приведена на рис.4.

Каждому модулю в системе присваивается по одной шине для запроса общей шины  $BR_i$  /Bus Request/ и для подтверждения запроса  $BG_i$  /Bus Granted/. Если какому-либо модулю необходимо занять ОШ для обмена информацией, то он должен включить свой  $BR_i$ . Этот модуль может подключаться к ОШ, если:

- он принял свой  $BG_i$  и
- закончена предыдущая операция с использованием ОШ /сигнал  $BUSY$  не включен/.

Когда новый модуль берет на себя управление ОШ, он должен включить сигнал  $BUSY$ . После выключения  $BR_i$  и  $BUSY$  текущий главный модуль освобождает ОШ. Сам арбитр не может прерывать работу модуля с шиной.

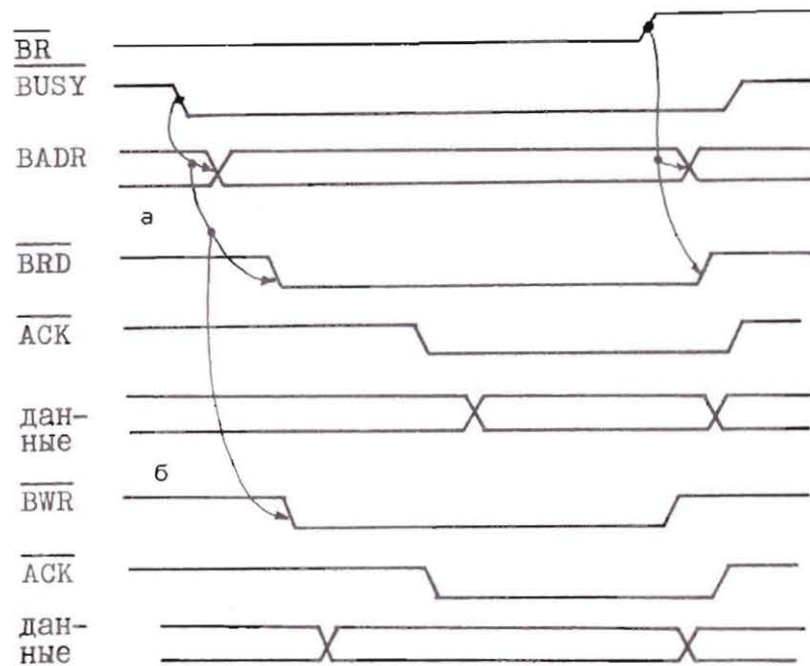


Рис.5. Операции на общей шине: а/ чтение, б/ запись.

Время захвата ОШ делится на интервалы фиксированной длины. Любой главный модуль может занимать ОШ на время, равное  $m$  таким интервалам / $m$  - целое число/, в зависимости от быстродействия как самого главного модуля, так и заказанного или подчиненного модуля. Этот полусинхронный способ имеет по сравнению с синхронным преимущество, заключающееся в лучшем согласовании скоростей выполнения операций в разнотипных модулях, а по сравнению с асинхронным способом, который обычно применяется в таких ситуациях, уменьшаются затраты на техническую реализацию.

Для обмена информацией ОШ имеет 27 шин, распределенных следующим образом:

- 16 адресных шин, позволяющих адресовать 64К ячеек памяти, регистров и портов ввода-вывода  $AB0...AB15$ ;
- 8 шин данных:  $DB0, \dots, DB7$ ;
- 3 шины управления:
 

BRD	- BUS READ	/чтение/
BWR	- BUS Write	/запись/
ACK	- Acknowledge	/сигнал подтверждения со стороны адресуемого модуля/.

На рис.5 приводится временная диаграмма выполнения операций через ОШ. В общем случае шина занимается только на время пере-



дачи одного слова /байта/. Для реализации непрерывной передачи блока данных необходимо включить сигнал BUSY на время передачи целого блока. Однако нужно учитывать, что на это время другие модули не будут иметь доступа к общей шине.

### 3.2.2. Межмодульные связи и синхронизация в системе

Модули ММПС обрабатывают отдельные части общей задачи системы. Для управления этим процессом составляются задания, которые передаются соответствующим модулям на исполнение. Эти задания содержат код операции и все необходимые параметры. В дальнейшем модуль, который составляет задание, будем называть заказчиком, а модуль, который выполняет задание, - исполнителем. Допускается, что модуль-исполнитель во время обработки полученного задания может выступать, в свою очередь, как заказчик по отношению к третьему модулю.

Передача заданий осуществляется путем записи сообщений, содержащих всю необходимую для выполнения задания информацию, в специально зарезервированную для этой цели область памяти. Эти области памяти, получившие в литературе название "почтовый ящик", обычно помещаются в общую память, так что все модули имеют доступ к ним [8].

В соответствии с [9] в данной ММПС используется модифицированный вариант почтовых ящиков, который позволяет уменьшить число обращений к ОШ при обмене сообщениями. С этой целью заданное адресное пространство из 64К адресов разбивается следующим образом [рис.6]:

- 0...15К - индивидуальное адресное пространство модуля;
- 15К...16К - адреса для почтовых ящиков;
- 16К...64К - адресное пространство общей памяти.

Схемная реализация межмодульных связей предусматривает включение в каждый модуль специального блока памяти /управляющая память/, имеющего двухканальный интерфейс /dual-port memory/. Доступ к управляющей памяти возможен как со стороны ОШ /адреса от 15К до 16К/, так и со стороны соответствующего модуля [рис.7]. Этот способ реализации почтовых ящиков дает следующие преимущества:

- при обмене сообщениями число обращений к ОШ уменьшается вдвое;
- чтение сообщения из блока управляющей памяти производится без обращения к ОШ;
- при записи сообщения в управляющую память со стороны ОШ схемным путем генерируется сигнал прерывания, поступающий на процессор внутри модуля, что позволяет немедленно инициировать обработку задания. Когда закончена обработка задания или произошел сбой при его обработке, необходимо известить об этом заказчика. Для этой цели в системе предусмотрены два механизма:

- статусные ячейки в управляющей памяти, которые могут опрашиваться заказчиком;
- специальные шины в ОШ, которые в указанных случаях включаются исполнителем, вызывая прерывание в модуле-заказчике.



Рис.6. Распределение адресного пространства в мульти-микропроцессорной системе.

Эти шины называются линиями запроса на обслуживание. В данной ММПС имеется 8 таких шин /SR0...SR7/, использование которых позволяет осуществлять быстрый обмен сообщениями в критических по времени случаях.

На рис.8 показана схема межмодульных связей ММПС.



Рис.7. Подключение модуля к общей шине.

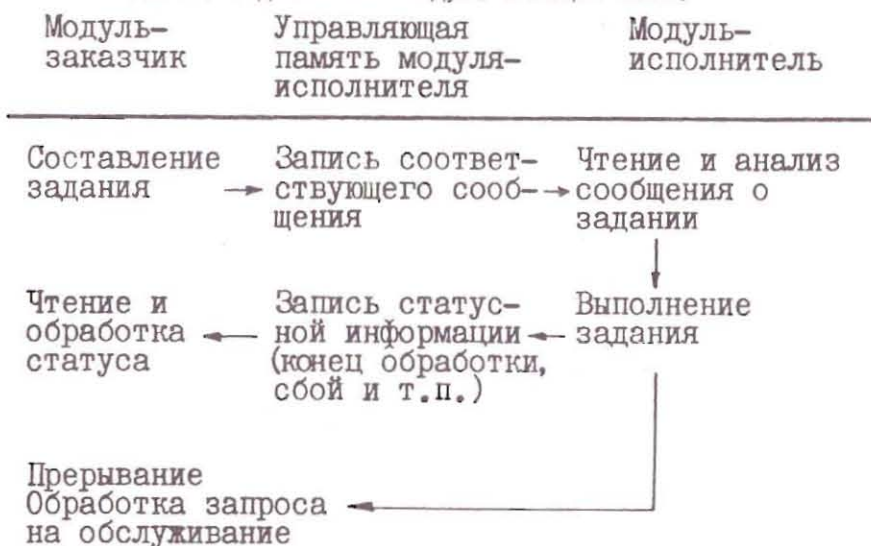


Рис.8

В ММПС допускается, что услугами одного модуля пользуются несколько других модулей, то есть модуль может стать общим ресурсом в системе. Типичный пример - это модуль общей памяти.

В данной системе арифметический модуль также является общим ресурсом, поскольку его вычислительные возможности могут использоваться процессором-монитором и дисплейным модулем.

Для управления доступом к общим ресурсам предусмотрены специальные флаги, реализующие операции опроса и установки /test and set - TAS/. Синхронизация обработки заданий во время использования ресурсов осуществляется при помощи BUSY-флагов. Указанные флаги реализованы схемным путем.

Допускается, что при захвате общего ресурса каким-либо модулем обрабатывается несколько заданий подряд /например, последовательность арифметических операций/. На рис.9 показан такой случай.

Описанные принципы межмодульной связи и синхронизации осуществляют слабую связь между процессами, протекающими параллельно в отдельных модулях. Это соответствует специфике применения ММПС для управления интеллектуальным терминалом, в котором параллельно обрабатываются следующие типы задач:

- диалог пользователь-терминал и/или центральная ЭВМ - терминал, организация и синхронизация обработки переданных заданий;
- преобразование графических объектов, а также их занесение в дисплейный файл;
- обработка дисплейного файла, то есть построение изображения на экране;
- некоторые вычисления по программе пользователя.

Схемная реализация флагов и выделение специальных шин облегчают межмодульные связи на программном уровне.

Наряду с описанным выше механизмом межмодульной связи в ММПС предусмотрена возможность быстрого обмена сообщениями /например, для обработки прерываний от светового карандаша/.

#### 4. ВНУТРЕННЯЯ АРХИТЕКТУРА МОДУЛЕЙ ММПС

##### 4.1. Варианты реализации

Имеющиеся в настоящее время наборы цифровых микроэлектронных компонентов позволяют реализовать следующие основные структуры, предназначенные для выполнения сложных логических и вычислительных функций:

- модуль на основе универсального 8-разрядного микропроцессора /МП/;
- модуль на основе 16-разрядного МП;
- специализированный процессор на базе микропроцессорных секций /ПС/;
- модуль с жесткой логикой на основе базовых интегральных схем /ИС/.

Модуль на основе универсального 8-разрядного МП требует наименьших затрат на реализацию, так как в настоящее время накоплен



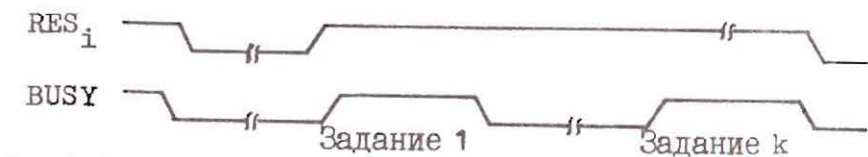
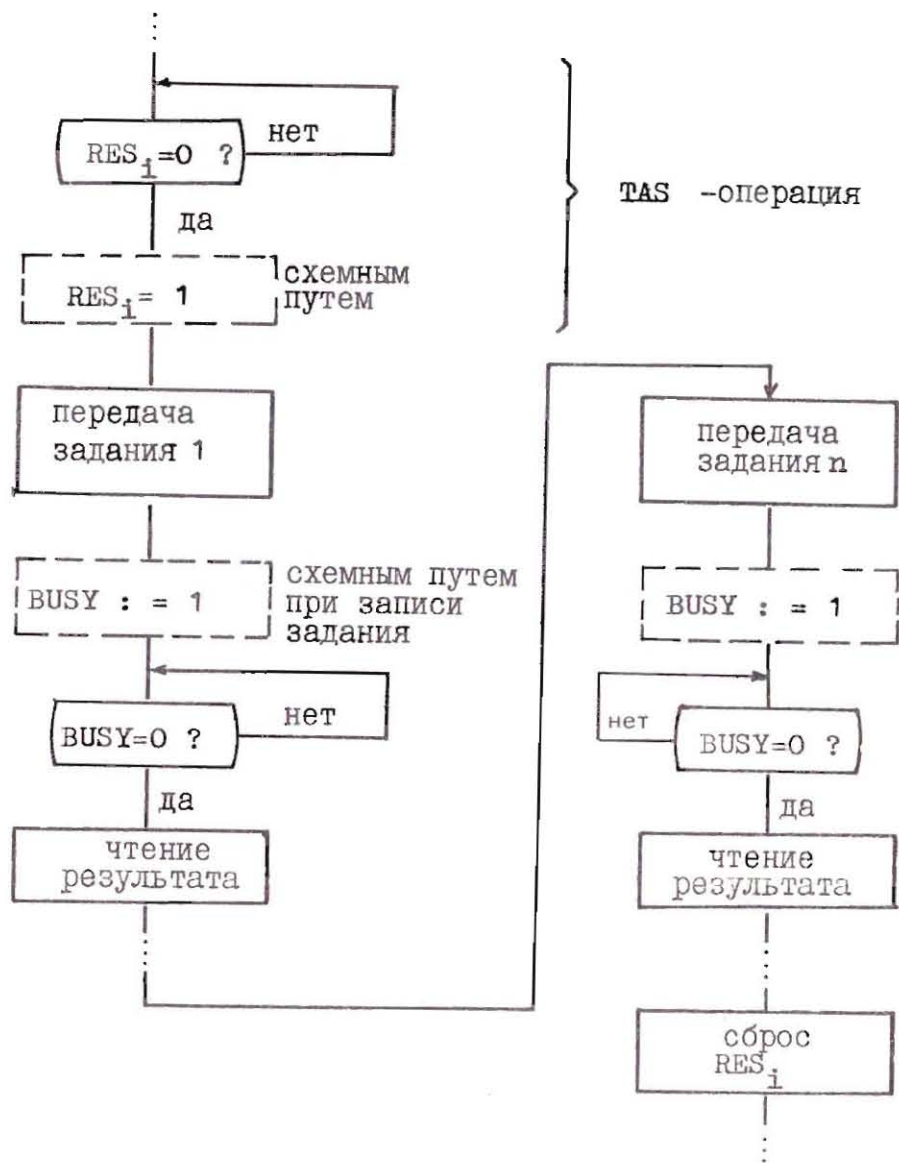


Рис.9. Синхронизация обработки заданий.  $RES_i$  - TAS-флаг для ресурса  $i$ ;  $BUSY$  - флаг для синхронизации / $BUSY=1$  - задание обрабатывается/.

большой опыт построения таких систем. Кроме того, существует большой фонд математического обеспечения для различных применений, а также хорошо оснащенные системы для разработки и отладки микро-ЭВМ/10/. Хотя быстродействие этих МП сравнительно велико /2,5÷5 мкс/команду/, их разрядность /8 бит/ для многих задач явно недостаточна, так как последовательная обработка информации с большей разрядностью приводит к значительному снижению производительности.

Новые 16-разрядные МП работают примерно в 2÷4 раза быстрее. Кроме того, благодаря большей разрядности, они имеют в десятки раз большую производительность, чем 8-разрядные МП/11/. И, наконец, еще одним важным преимуществом, обусловившим широкое распространение 16-разрядных МП, является более совершенная внутренняя архитектура /расширенный список команд, включая умножение, деление и др.; введение дополнительных способов адресации; увеличение адресуемой памяти до 1÷16 Мбайт/.

Микропроцессорные секции появились в результате развития биполярных технологий ТТЛ и ЭСЛ. В настоящее время серийно выпускаются процессорные секции, секции микропрограммных контроллеров, контроллеров ввода-вывода и др./12,13/. Таким образом, стало возможным при использовании небольшого числа ИС строить мощные микропрограммируемые процессоры с подходящей для конкретных применений архитектурой. Разработка процессоров на основе секций требует более глубоких знаний в области архитектуры ЭВМ и микропрограммирования и связана с более высокими затратами по сравнению с использованием универсальных МП. Приобретение же специальных систем для разработки микро-ЭВМ целесообразно только тогда, когда разрабатывается большое число систем на базе таких компонентов.

Процессоры с жесткой логикой, построенные на быстрых биполярных ИС малой и средней интеграции, имеют самое высокое быстродействие среди четырех рассмотренных типов процессоров. Они применяются тогда, когда требуемое быстродействие не может быть достигнуто другими средствами. Очевидно, что их реализация требует значительных затрат, а внесение изменений в законченную разработку чрезвычайно трудоемко.

Рассмотрим возможность использования в ММПС описанных выше структур на примере дисплейного модуля, предназначенного в первую очередь для построения графических примитивов на экране векторного дисплея, в частности линий и символов. Для анализа выберем основной параметр - время построения указанных графических элементов. Укажем также, что символы строятся из коротких отрезков, а для построения линий используется быстрый алгоритм Брезенхама/14/. Вычисления времени построения линий и символов были сделаны для широко распространенных универсальных 8- и 16-разрядных МП I8080 /КР580ИК80/, I8086, процессорных секций I3002 /К589ИКО2/ и интегральных схем SN74S181 /К531ИПЗ/. Результаты вычислений приведены в табл.1, причем в указанных вариантах опера-

ции построения осуществляются программой, микропрограммой и жесткой логикой соответственно.

Таблица 1

Вариант	Время построения в мс		Скорость построения	
	Линии (1024 точки)	Символы ("А")	Линий в с	Символов в с
8-разряд- ный МП	253	16,7	4	60
16-разряд- ный МП	81,5	3,6	12,5	287
Процессор- ные секции	1,45	0,07	690	14285
Процессор с жесткой логикой	0,07	0,004	14285	250000

Примечание: Указанные в таблице времена не учитывают загрузку параметров из памяти. Программы для первых двух вариантов написаны на языке PL/M80 и PL/M86 соответственно.

Из табл.1 видно, что только последние два варианта реализации генератора линий приемлемы для построения мощного дисплея векторного типа.

#### 4.2. Структура типичного модуля

Для программы построения линий в случае использования 8-разрядного МП в табл.2 сопоставлены длина и время выполнения подготовительной и исполнительной частей программы /для линии, состоящей из 1024 точек/. При этом рассматриваются три варианта:

- I - простая версия программы по алгоритму Брезенхама;
- II - составление программы для каждого из октантов /таким образом уменьшается число условий, которые обрабатываются в цикле/;
- III - оптимизация тех участков программы, которые обрабатываются в цикле.

Из этой таблицы следует, что при построении линий примерно 92÷99% общего времени тратится на обработку небольшой части

Таблица 2

Вариант	Подготовка (постоянная часть)		Выполнение (переменная часть)	
	длина програм- мы, %	время выпол- нения, мс	длина програм- мы, %	время выпол- нения, мс
I	78,4	0,58	21,6	252,4
II	92,8	0,67	7,2	191
III	95,8	0,67	4,2	90,6

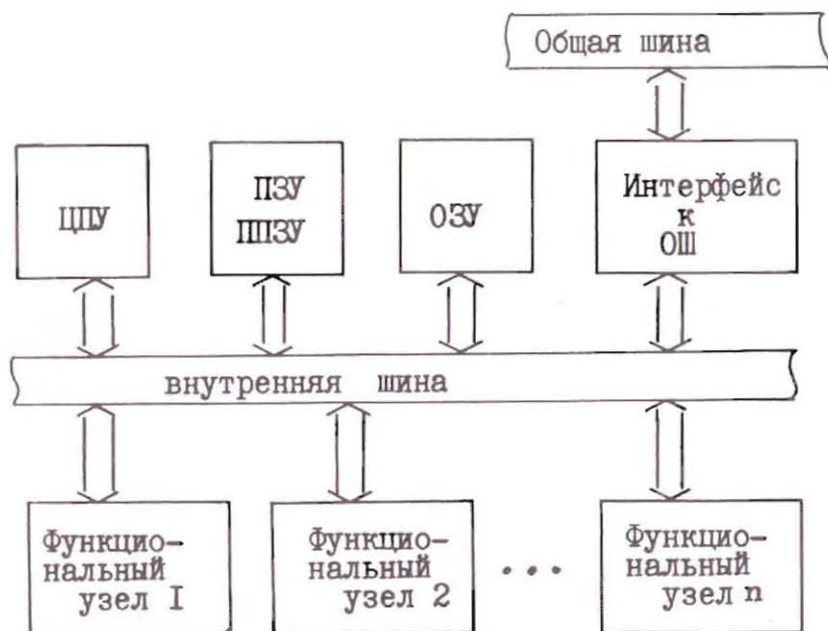


Рис.10. Структура модуля ММПС.

/4÷22% / всей программы. С другой стороны, анализ задания и его подготовка к исполнению составляют основную часть программы, но почти не влияют на время построения линии. Можно показать, что такая ситуация типична и для многих других задач.

Итак, если при технической реализации функционального модуля заданные требования к быстродействию могут быть выполнены только путем применения биполярного быстрого процессора, то целесообразно



но передать спецпроцессору на исполнение не весь комплекс функций данного модуля, а только ту часть, которая в основном определяет время выполнения всей задачи. Для выполнения остальных функций следует использовать универсальный МП. Структура такого модуля показана на рис.10. Он состоит из стандартной микро-ЭВМ /МП, память, прерывание и др./ и из одного или более функциональных узлов, которые управляются микропроцессором.

Распределение задач внутри модуля осуществляется следующим образом:

- микропроцессор - инициализация модуля;
- связь с другими модулями ММПС;
- прием и подтверждение заданий;
- подготовка и синхронизация обработки заданий;
- обработка ошибок.

Функциональные узлы - быстрая обработка специализированных задач.

Построенный по такому принципу дисплейный модуль обеспечивает время построения линии 130 мкс, а символа -  $2 \times 8$  мкс, что сравнимо с временами, указанными в табл.1 для схем с жесткой логикой.

Рассмотренная архитектура по сравнению с чисто микропрограммным или с жестко запаянным спецпроцессором имеет следующие преимущества:

1. Уменьшается доля схем специального назначения и, таким образом, уменьшаются аппаратные затраты.
2. Облегчается разработка управляющих программ, так как основная их часть может быть написана на ассемблере МП или на языке высокого уровня.
3. Внутримодульный МП можно использовать при отладке для тестирования и проверки функциональных узлов.
4. Структура универсальной микро-ЭВМ повторяется от модуля к модулю и тем самым достигается большая степень унификации схемных решений.

Связь и синхронизация внутри модуля осуществляется по тем же принципам, что и на уровне ММПС в целом.

#### 4.3. Структура функционального узла

В составе модуля функциональный узел выполняет узко ограниченный комплекс задач. Если нет готовой специализированной ИС с быстройдействием, достаточным для решения данной конкретной задачи, то такой узел может быть построен на основе микропрограммируемого секционного процессора или на базе схем с жесткой логикой. При этом следует отметить, что микропрограммируемый процессор имеет большую степень гибкости и обеспечивает большую регулярность схемной реализации.

В тех случаях, когда быстроедействие секционного процессора недостаточно, функциональный узел можно строить так же, как

и функциональный модуль: микропрограммируемый спецпроцессор управляет при помощи внутренней шины некоторым набором блоков с жесткой логикой. Чтобы уменьшить затраты и достичь определенной степени гибкости, эти блоки реализуются на основе программируемых логических матриц /ПЛИМ/ или на быстрых ПЗУ.

В заключение в табл.3 показаны основные архитектурные уровни разработанной ММПС, а также элементы их схемной реализации.

Таблица 3

Уровень	Элементы схемной реализации	Реализация функций	Связи
ММПС	Функциональные модули	Программы на языке высокого уровня	МП ↔ МП
Модуль	Микропроцессоры	Программы на ассемблере	МП ↔ ПС
Функциональный узел	Микропроцессорные секции	Микропрограммы	ПС ↔ специализированный блок
Подсистемы		ИС, ПЛИМ, ПЗУ	-

Рассмотренный подход к проектированию модулей ММПС позволил получить высокое быстродействие и дал значительный выигрыш во времени и затратах на разработку.

Вопросы технической реализации ММПС будут детально рассмотрены в следующей работе.

#### ЛИТЕРАТУРА

1. French H. Computer Design, 1981, vol.20, No.7, p.99.
2. Левчановский Ф.В. ОИЯИ, 11-81-497, Дубна, 1981.
3. Leich H. et al. JINR, E11-81-296, Dubna, 1981.
4. Draft International Standard, ISO/TC97/SC5/WG2, Information Processing: Graphical Kernel System (GKS), Functional Description, 1980.
5. Newman W.M., Sproull R.F. Principles of Interactive Computer Graphics. McGraw-Hill, New York, 1973.
6. Leich H., Levchanovsky F.V. JINR, E11-81-297, Dubna, 1981.
7. Hoener S., Roender W. In: Proc. of the Third Euromicro Symposium on Microprocessors and Microprogramming. Amsterdam, October 1977, p.35.

8. Ford W.S., Hamacher V.C. In: Proc. of the COMPCON. Fall, 1976, p.113.
9. Hirschmann A.D. et al. Computer Design, 1979, vol.18, No.5, p.181.
10. Klingman E.E. Microprocessor System Design. Prentice-Hall, Inc., New Jersey, 1977.
11. Katz B.J. et al. Electronics, 1978, No.4, p.99.
12. Bipolar Microprocessor Logic and Interface Data Book. Advanced Microdevices, Inc., 1981.
13. Березенко А.И. и др. Микропроцессорные комплекты повышенного быстродействия. "Радио и связь", М., 1981.
14. Bresenham J.E. IBM Systems Journal, 1965, vol.4, No.1, p.25.

Лайх Х., Левчановский Ф.В., Приходько В.И. P10-83-7  
 Мультимикропроцессорная система управления интеллектуальным графическим терминалом.  
 Архитектура системы

Рассматриваются структура и принципы построения мультимикропроцессорной системы, предназначенной для управления интеллектуальным графическим терминалом. Высокая производительность системы при низких затратах достигается путем построения функциональных модулей на основе универсальных микропроцессоров, микропроцессорных секций и схем с жесткой логикой, связанных в единый комплекс.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1983

Leich H., Levchanovsky F.V., Prikhodko V.I. P10-83-7  
 Multi-Microprocessor System for Control of an Intelligent Graphic Terminal.  
 System Architecture

Architecture and design principles of a multi-microprocessor system intended for control of an intelligent graphic terminal are described. High system performance paired with low cost is obtained by combining microprocessor, bit-slice-elements and hardware logic in the module building blocks.

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

Preprint of the Joint Institute for Nuclear Research. Dubna 1983

Рукопись поступила в издательский отдел  
 7 января 1983 года.

Перевод О.С.Виноградовой.