ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ
СО АН СССР

М.В.Бейлин, О.В.Вьюшин,
А.Д.Клименко, В.В.Конюхов,
В.Ф.Кузнецов, А.В.Проскурин,
П.Л.Храпкин, А.К.Эпштейн

СИСТЕМНОЕ МАТОБЕСПЕЧЕНИЕ МНОГОМАШИННОГО КОМПЛЕКСА АВТОМАТИЗАЦИИ ФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА

ПРЕПРИНТ 82-72

СИСТЕМНОЕ МАТОБЕСПЕЧЕНИЕ МНОГОМАШИННОГО КОМПЛЕКСА АВТОМАТИЗАЦИИ ФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА

Бейлин М.В., Вьюшин О.В., Клименко А.Д.,
Конюхов В.В., Кузнецов В.Ф., Проскурин А.В.,
Храпкин П.Л., Эпштейн А.К.

АННОТАЦИЯ

Описывается системное матобеспечение многомашинного комплекса автоматизации физического эксперимента, построенного на базе мини-ЭВМ "Электроника-100/25" и "СМ-4", а также микро-ЭВМ "Электроника-60". Комплекс имеет радиальную структуру, состоящую из одной мини-ЭВМ в качестве центральной машины и до пяти микро-ЭВМ на периферии. Системное матобеспечение ориентировано на работу в режиме реального времени и построено на базе операционной системы RSXIIM. В качестве интерфейса связи, используется разработанный в ИЯФ интерфейс DL-Ki/Si. Матобеспечение включает развитую систему связи, ориентированную на использование периферийных микро-ЭВМ как в качестве рабочих мест экспериментатора, так для управления экспериментальными установками. В качестве основного языка для периферийной ЭВМ используется ФОРТРАН. В состав комплекса входит развитая библиотека для работы с КАМАКОМ.

Г Л А В А I

ЦЕЛИ И ТРЕБОВАНИЯ

К моменту написания настоящего препринта в Институте ядерной физики СО АН СССР работало две системы автоматизации физического эксперимента. Регистрационно-управляющая система "Радиус", описанная в /2/ и состоящая из двадцати мини-ЭВМ "М-6000" и система "Одра", описанная в /1/ и состоящая из пятнадцати ЭВМ серии "Одра-1325". "Одра" представляет собой систему управления крупными экспериментальными установками. Особенности развития и становления этой системы позволили достичь уровня сервиса, дающего возможность быстрой и эффективной подготовки рабочих программ управления установкой. Простота системы и удобный интерфейс с пользователем, а также транслятор с языка высокого уровня типа "Фортран" /12/, написанный с учетом требований пользователя и специфики работы, сделали эту систему одной из распространенных в Институте.

На базе нового поколения ЭВМ типа "Электроника" и СМ-4 была начата разработка многомашинного комплекса автоматизации физического эксперимента. В качестве основного было выбрано направление автоматизации установок, что, с одной стороны, обусловило выбор операционных систем, а с другой стороны, определило конфигурацию комплекса. Каждая установка располагает центральной ЭВМ типа СМ-4 или Электроника 100/25 и несколькими периферийными микро-ЭВМ типа Электроника-60. В настоящий момент работают десять таких комплексов и на их основе создается локальная сеть с общей архивной машиной.

Ориентация комплекса на обслуживание установок означает необходимость применения многопрограммного режима в центральной ЭВМ, а также подготовку и загрузку программ в переферийную ЭВМ. Кроме того, необходимы средства синхронизации программ в центральной и периферийных машинах. Поэтому в качестве операционной системы для центральной ЭВМ использована ОС реального времени RSX-11M, которая имеет развитые средства мультипрограммирования и многопользовательской защиты.

Вместе с тем, применение многопользовательской ОС требует использования специального системного процесса для диспетчеризации доступа к аппаратуре, так называемого драйвера. Это, в свою

очередь, означает, что на оформление каждого запроса уходит время порядка 2-I0 миллисекунд. Причем это время слабо зависит от производительности ЭВМ. С другой стороны, в однопользовательских системах нет необходимости использовать драйверы, что дает уменьшение времени реакции и оформления запроса примерно на порядок. Таким образом, периферийная ЭВМ дает возможность не только значительно увеличить эффективность работы с аппаратурой, но и повысить производительность процессора центральной ЭВМ.

Важным элементом системы управления является язык высокого уровня для написания программ управления. Таким языком для большинства установок института является Фортран. Использование Фортрана в периферийной ЭВМ требует наличия в ней операционной системы. С целью упрощения интерфейса с пользователем в качестве операционной системы для переферийной ЭВМ была выбрана операционная система RSX IIS , полностью совместимая с ОС центральной машины.

В процессе создания комплекса выяснилось, что появление микро-ЭВМ "Электроника-60" привело к появлению большого числа пользователей, для которых потребность в средствах вычислительной техники полностью удовлетворяется одной микро-ЭВМ. Сюда относятся локальные рабочие места физиков-экспериментаторов (так называемые мэксы), имеющие один - три крейта КАМАК, а также рабочие места инженеров-электронщиков. Для этого класса пользователей существенным является многопрограммность в периферийной ЭВМ, средства защиты пользователей от взаимного влияния в центре, а также возможность подготовки программ с терминала микро-ЭВМ и необходимость записывать информацию на диски центральной машины. Анализ требований, предъявляемых к комплексу с точки зрения управления установками и обслуживания мэксов, показывает, что эти требования в основном совпадают. Таким образом, возможно использование единой конфигурации для организации управления как установками, так и мэксами.

Поскольку в данный момент подавляющее большинство пользователей работают в режиме мэкса, была предпринята попытка создать вариант системы, наиболее подходящий для таких пользователей. Это привело к появлению первого этапа системы связи. Он состоит из ОС RSX IIM в центре, системы начальной загрузки и ОС RSX IIS на периферии. Однако первый этап системы не обеспечивает многопрограммности на уровне Фортрана в микро-ЭВМ, а также не дает возможности записывать информацию на диски из программ в периферийной ЭВМ. Для решения этих проблем был реализован второй этап системы, который является расширением первого, и использует его в качестве вспомогательной системы.

Этот этап включает в себя:

I. Драйвер линии связи.
2. Средства межпрограммного обмена.
3. Загрузку программ с дисков центральной ЭВМ.
4. Обеспечение доступа к удаленным файлам.
5. Виртуальные терминалы.

Архитектурная схема комплекса представлена на рисунке I. Для пользователя комплекс обеспечивает следующее:

I. ЭВМ приближается к аппаратуре - при прочих равных условиях это дает увеличение скорости обмена с аппаратурой в 5-I0 раз.
2. У пользователя появляется собственная ЭВМ - если учесть, что подготовку программ он может вести в центральной машине, в то время как его программы выполняются в переферийной ЭВМ, то это означает увеличение производительности в I,5-2 раза по сравнению с использованием только центральной ЭВМ для этих целей.
3. Возможность объединения комплексов в локальную сеть, содержащую централизованный архив.

Все эти особенности проявляются как при использовании комплекса для управления установкой, так и при обслуживании рабочих мест. Кроме того, есть еще один важный аспект развития комплекса. Дело в том, что со временем некоторые мэксы перерастают в установки. Такое изменение организации управления не должно приводить к несовместимости с уже написанным матобеспечением. Описанная выше структура комплекса позволяет не только сохранить совместимость, но и с успехом использовать уже написанное пользователем матобеспечение при переходе на новый тип организации управления.

Г Л А В А 2

ВЗАИМОДЕЙСТВИЕ ПОЛЬЗОВАТЕЛЯ С СИСТЕМОЙ

2.I. Процедура начальной загрузки.

После включения питания периферийной ЭВМ, она переходит в режим удаленного терминала. Перед началом работы необходимо загрузить операционную систему в периферийную ЭВМ и выполнить операции включения системы связи. Для этого нужно представиться системе в центральной машине и выдать команду начальной загрузки. Команда начальной загрузки имеет следующий формат:

LDS TTN:=DKN:[g,N]RSXIIS.SYS <CMD1> ... <CMDN>

где:

LDS – название команды начальной загрузки,

TTN: – терминал, который представляет периферийную ЭВМ,

DKN:[g,n]RSXIIS.SYS – координаты файла операционной системы, которая будет загружена в периферийную ЭВМ,

<CMD1> ... <CMDN> – необязательные параметры, представляющие собой команды операционной системы в центральной ЭВМ, которые будут выполнены после того, как периферийная ЭВМ будет загружена и инициирована линия связи.

В соответствии с принятыми в системе RS X II умолчаниями, если параметры команды не указаны, она выглядит следующим образом:

LDS TI:=SY:[1,60] RSXIIS.SYS

где: TI: – терминал с которого выдана команда, а SY: – диск пользователя. Поэтому командная строка для выполнения начальной загрузки с терминала периферийной ЭВМ, может быть сокращена до:

LDS =

Вместо опущенных параметров будут автоматически подставлены значения по умолчанию, т.е. строки: LDS TI:=SY:[1,60]RSXIIS.SYS и LDS = полностью эквивалентны. Если же необходимо запустить одну или несколько программ, которые работают с линией связи, то команда начальной загрузки будет выглядеть так:

LDS = <RUN PROG1> <RUN PROG2> . ..

где PROG1, PROG2 – программы пользователя, работающие с линией связи.

2.2. Программа связи с оператором в периферийной ЭВМ.

После выполнения начальной загрузки периферийной ЭВМ, с терминалом пользователя работает операционная система RS x IIS. Интерпретатором команд пользователя в системе RS X II является специальная программа, которая называется MCR. Для системы RSxIIS был написан свой вариант такой программы, что позволило сэкономить примерно 2 К слов оперативной памяти. В отличие от MCR в центральной ЭВМ, в эту программу включены только наиболее необходимые команды, а также ряд команд, не входящих в набор MCR в центральной ЭВМ. Команды MCR для периферийной машины описаны ниже:

1. RUN PROG – запуск программы
2. ABO PROG – прекращение выполнения активной программы
3. RES PROG – продолжение выполнения приостановленной программы
4. REM PROG – исключение программы из списка доступных для запуска
5. INS PROG* – включение программы в список доступных для запуска
6. LUN PROG* – вывод списка логических номеров устройств программы
7. REA*PROG – переназначение логических номеров устройств программы
8. LIN* – вывод состояния линии связи
9. ACT – вывод списка активных программ в системе
10. TAS – вывод списка программ, доступных для запуска
11. POOL – вывод состояния системного пула
12. BOOT – переход в режим начального загрузчика
13. RMT* – переход в режим удаленного терминала
14. LCT* – возврат в режим местного терминала

Звездочкой помечены команды, реализованные в виде отдельных программ (расширенные директивы).

2.3. Межпрограммный обмен по линии связи.

Для обмена данными и синхронизирующей информацией между программами в центральной и периферийной ЭВМ существует пакет подпрограмм. Пакет используется при программировании на Фортране и является общим для программ центра и периферии.

Всего существует пять подпрограмм, осуществляющих межпрограммный обмен по линии связи и выполняющих следующие функции:

XOPEN - прикрепление одного из 8-ми логических каналов линии связи и заполнение блока описания состояния канала;

XCLOSE - отключение логического канала и зануление соответствующего блока описания состояния канала;

XPINT - передача синхронизирующей информации;

XREAD - чтение информации по указанному логическому каналу;

XWRITE - запись информации по указанному логическому каналу.

Обращение к программам соответствует принятому в Фортране:

```
CALL NAME (BI,...,BN)
```

где NAME - имя подпрограммы, а BI,...,BN - список параметров.

Названия, список и функции параметров подпрограмм пакета следующие:

```
XOPEN(  LUN , K  SYNFL,  AST ,  ISYN , IC )
XCLOSE ( LUN ,  IC ),
XPINT ( LUN , CODE,  IC ),
XREAD (  LUN , ADDR, NWR,  NWW ,  IC ),
XWRITE(LUN , ADDR, NWW,  NWR ,  IS   IC )
```

Параметры, заключенные в квадратные скобки - необязательны.

LUN - логический номер устройства, назначенный на используемый канал драйвера линии связи. Изменяется в пределах от I до 255. По умолчанию LUN = 1. Если в подпрограммах XPINT , XCLOSE , XREAD , XWRITE , используется значение LUN не соответствующее уже открытому каналу, то на терминал выводится сообщение об ошибке - ERROR CODE 32 (INVALID LUN) и задача завершается.

K - номер подключаемого канала драйвера линии связи. По умолчанию K=2.

SYNFL - номер флага для синхронизации обмена между программами-партнерами в разных машинах. Изменяется в пределах от 9 до I6.

AST - адрес программы обработки прерывания, которое возникает при получении синхросигнала от программы-партнера. Этот адрес предварительно описывается в операторе EXTERNAL.

CODE - код синхросигнала, передаваемого в другую машину. Изменяется в пределах от I до "376.

ISYN - параметр, определяющий, что коды Ø и "377, получаемые при старте и завершении программы-партнера, должны передаваться в подпрограмму обработки прерываний данной задачи. Если ISYN не указан, то коды Ø и "377 в программу не передаются.

ADDR - адрес массива (INTEGER*2) для приема или для передачи.

NWR - для подпрограммы XREAD этот счетчик слов, подлежащих приему. Для подпрограммы XWRITE - счетчик в действительности переданных слов (выходной параметр).

NWW - для подпрограммы XREAD это счетчик в действительности принятых слов (выходной параметр). Для подпрограммы XWRITE - счетчик слов, подлежащих передаче.

IS - значение временного интервала (в миллисекундах). В течение этого интервала перед выполнением функции записи происходит ожидание функции чтения от программы-партнера.

IC - статус завершения подпрограммы. Принимает следующие значения:

IC=1 - OK

если IC не равно I · В IC код ошибки в виде восьмеричного числа.

Во всех подпрограммах пакета используется одинаковое соглашение относительно параметра IC. Если статус завершения указан, то восьмеричный код регистрируемой ошибки просто отдается в качестве значения этого параметра, если не указан - на терминал выводится сообщение соответствующие обычным Фортрановским ошибкам. Если эта ошибка фатальная - выполнение программы завершается, если нет, то после вывода сообщения, выполнение программы продолжается. Смысл использования параметра IC заключается в том, чтобы при анализе IC распознать ошибку, которая может произойти во время выполнения программы. В соответствии с этим при программировании предусматриваются определенные действия.

2.4. Пакет подпрограмм для работы с терминалом.

Существующие в стандартном ФОРТРАНЕ средства для работы с терминалом ориентированы на работу с печатающими устройствами и не используют возможностей дисплеев. Кроме того, они не приспособлены для диалоговой работы и ограничены по возможностям. Все это делает их неудобными для пользователя. Большой опыт эксплуатации системы "ОДРА" говорит о том, что хорошо развитые средства для работы с терминалом являются важным компонентом управляющего комплекса.

Эти обстоятельства привели к созданию специального пакета для работы с терминалом, внешний интерфейс которого был заимствован у соответствующего пакета в системе "ОДРА". Пакет ориентирован на работу с алфавитно-цифровыми дисплеями типа Видеотон-340 и им подобными. Он включает большое количество сервисных средств: редактирование вводимой строки, вывод предыдущего значения строки для редактирования, обширный и задаваемый пользователем набор терминаторов, т.е. символов завершающих ввод строки, автоматическое распознавание директив, переключение режимов ввода/вывода, средства управления экраном, позиционирование курсора и т.п. Кроме того, пакет обеспечивает ввод информации в асинхронном режиме, что позволяет вести диалог с несколькими программами, динамически переключать режимы работы, а также контролировать обращения к системе. Пакет может использоваться для работы как в центральной, так и в периферийной ЭВМ.

2.4.I. Редактор в периферийной ЭВМ.

При числе пользователей комплекса от трех и более проблема памяти в центральной ЭВМ становится все более острой, особенно для редактирования текста, а увеличение времени реакции на принятый в редакторе символ резко уменьшает степень комфорта при работе с ним. Это привело к необходимости перенести редактор в периферийную машину.

В качестве исходного был использован редактор текстов, написанный в Институте Автоматики и Электрометрии СО АН СССР.Он был модифицирован для работы в периферийной ЭВМ с использованием средств доступа к файлам на дисках центральной машины. Это позволило вести одновременное редактирование пяти и более пользователям, не мешая друг другу, тогда как редактор в центральной ЭВМ дает такую возможность только для троих. Помимо этого, редактор в периферийной машине имеет в два-три раза меньше время реакции по сравнению с редактором, работающим с виртуальным терминалом.

2.5. Доступ к файлам центральной ЭВМ.

Для обеспечения доступа к файлам центральной ЭВМ из периферийной машины написан специальный пакет подпрограмм. Этот пакет используется программами, написанными на Фортране, и предназначенными для запуска в периферийной ЭВМ.

Открытие файла, находящегося на любом из внешних запоминающих устройств центральной ЭВМ, выполняется оператором Фортрана OPEN . Смысл и внешний вид параметров оператора OPEN для программ в периферийной ЭВМ сохраняется. Исключение составляет параметр UNIT , который определяет, в данном случае, не номер логического устройства, а номер подканала линии связи, по которому предполагается вести обмен данными.

Закрытие файла производится стандартным оператором Фортрана C L O S E.Параметр UNIT в операторе должен совпадать с аналогичным параметром оператора OPEN , выполнявшегося для того же файла.

. Чтение данных; так же как и запись, выполняется блоками длиной не более I6K слов. Для чтения данных из файлов центральной ЭВМ предназначена подпрограмма FREAD. Формат обращения из программы, написанной на Фортране:

CALL FREAD (IUNIT, IADDR, ISIZE, [IVBN, IERR])

IUNIT - номер логического канала, по которому ведется обмен информацией.

IADDR - адрес массива, в который помещаются прочитанные из файла данные.

ISIZE - размер передаваемого блока.

IVBN - номер 5I2-ти байтового виртуального блока в файле, начиная с которого ведется чтение данных.

IERR - переменная, типа INTEGER*2 , содержит статус завершения подпрограммы. Коды ошибок, представленные младшим байтом, соответствуют стандартным кодам ошибок ввода/вывода и кодам ошибок директив, принятым в операционной системе RSXII (смотри 11).

Возможно возникновение ошибок трех уровней:

- значение переменной IERR находится в пределах I28-256 - ошибки в фактических параметрах вызова.
- значение переменной IERR больше 32768.

(4ØØØØB) – ошибка передачи – сбои при передаче данных или служебной информации по линии связи.

– Значение переменной IERR меньше Ø – файловые ошибки – сбои при выполнении файловых операций.

Первые три параметра подпрограммы являются обязательными. Если необязательный параметр IVBN не указан, чтение будет производиться, начиная со следующего за последним обработанным в предыдущей файловой операции блоком.

Для записи информации из определенного пользователем массива в файл центральной ЭВМ используется подпрограмма FWRITE Формат вызова подпрограммы из Фортрана:

CALL FWRITE(IUNIT,IADDR,ISIZE,[IVBN, IERR])

Все параметры имеют тот же смысл, что и в подпрограмме FREAD

2.6. Фортран в удаленной машине.

Как показывает опыт, большая часть прикладных программ в нашем Институте и во многих других научных центрах пишется на языке Фортран. Этот язык знаком подавляющему большинству физиков и инженеров. Составить и отладить программы управления экспериментом на Фортране проще и быстрее, чем на ассемблере. К преимуществам Фортрана следует также отнести наличие большого количества библиотек стандартных подпрограмм как в области численных методов, так и для прикладных применений.

В ИЯФ используются библиотеки научных подпрограмм фирмы DEC-SSP (scientific subroutines package) – пакет подпрограмм для численных расчетов, LSP (laboratory subroutines package) – набор подпрограмм, используемых при обработке данных, в частности, построение и обработка гистограмм, статистический и фурье – анализ, а также ряд описанных в этой работе пакетов, созданных в нашем Институте.

2.6.I. Методы отладки программ на Фортране,

Несмотря на то, что Фортран, как и другие языки программирования, позволяет обнаружить множество ошибок на стадии трансляции и обеспечивает диагностику при возникновении ошибок во время исполнения программы, при отладке больших и сложных программ полезно использовать специальный отладчик FDT –

– FORTRAN DEBUGGING TOOL. FDT занимает около 3-х К слов оперативной памяти и несколько замедляет исполнение проверяемых подпрограмм, но позволяет, не перетранслируя программу, динамически изменять и проверять значения переменных, прослеживать ход исполнения алгоритмов.

2.7. Работа с КАМАКОМ в периферийной машине.

Для работы с электронной аппаратурой в стандарте КАМАК был разработан пакет подпрограмм, являющийся реализацией международного стандарта ESONE/SR/01 9,10 . Подпрограммы пакета могут использовать контроллеры крейтов типа К-I6, Э-6Ø, М-4ØØ, СМ-3, СС-II и аналогичные. Возможна динамическая настройка пакета на конкретный тип контроллера, что обеспечивает мобильность программ, либо настройка в момент генерации пакета, позволяющая несколько сократить размеры подпрограмм и время их исполнения.

Подпрограммы пакета можно подразделить на следующие группы:

I. Декларативные подпрограммы, формирующие внутренние структуры данных пакета, например, декларации запросов на внимание и т.н. КАМАК-адресов (КАМАК-адрес состоит из номера крейта, позиции вставки в крейте N , субадреса A).

2. Подпрограммы общего управления, генерирующие команды КАМАК.

3. Подпрограммы, выполняющие единичные КАМАК-циклы с 8, I6 или 24-битовыми данными.

4. Подпрограммы, передающие целые массивы данных. Эти подпрограммы генерируют последовательность команд КАМАК, обеспечивая скорость обмена данными до 4Ø К слов/сек.

5. Подпрограммы, обслуживающие запросы на внимание – L. Существует 3 способа использования L : приостановка программы до появления L, при появлении L обслуживание запроса в режиме прерывания и проверка наличия L для организации ветвления в алгоритме.

Ниже приведен пример программы, использующей пакет для работы с КАМАК. Эта программа осуществляет проверку запоминающего устройства (ЗУ) емкостью 4Ø96 десятибитовых слов.

```
      INTEGER STAT,   !КАМАК-АДРЕС СТАТУСНОГО РЕГИСТРА ЗУ
     *          ADR,   !КАМАК-АДРЕС РЕГИСТРА АДРЕСА ЗУ
     *          ZU,    !КАМАК-АДРЕС РЕГИСТРА ДАННЫХ ЗУ
     *       BUF(4096), !БУФЕР ДЛЯ ДАННЫХ
     *        ICB(4)
      DATA ICB /4096,0,0,0/
C
C ДЕКЛАРАЦИЯ КАМАК-АДРЕСОВ
      CALL CDREG(STAT,,1,12,1)   !КРЕЙТ 1, N 12 A1
      CALL CDREG( ADR,,1,12,2)
      CALL CDREG(  ZU,,1,12)
C
C ЗАПОЛНЕНИЕ БУФЕРА СЛУЧАЙНЫМИ ЧИСЛАМИ
      DO 1 I=1,4096
      CALL RANDU(I1,I2,X)
1     BUF(I)=X*1023.
C
C ЗАПИСЬ В ЗУ
      CALL CCCC(ZU)          !КОМАНДА С
      CALL CSSA(17,STAT,3) !ЗАПИСЬ В СТАТ.РЕГИСТР РЕЖИМА 3
      CALL CSSA(17,ADR,0)  !НАЧАЛЬНЫЙ АДРЕС ЗУ 0
      CALL CSUBC(16,ZU,BUF,ICB) !ЗАПИСЬ В ЗУ МАССИВА BUF
      IF(ICB(2).LT.4095) STOP 'WRITE FAILURE'
C
C ЧТЕНИЕ ЗУ СО СРАВНЕНИЕМ
      CALL CSSA(17,STAT,1) !ЗАПИСЬ В СТАТ.РЕГИСТР РЕЖИМА 1
      CALL CSSA(17,ADR,0)  !НАЧАЛЬНЫЙ АДРЕС ЗУ 0
      DO 2 I=1,4096
      CALL CSSA(0,ZU,K)
      IF(I.EQ.1)GOTO 2
      IF(K.NE.BUF(I)) PAUSE 'COMPARE ERROR'
2     CONTINUE
      STOP 'OK'
      END
```

Приведенный пример демонстрирует передачу массива информации подпрограммой CSUBS , которая за 1ØØ мсек заполняет 4Ø96 слов ЗУ случайными числами. Считывание информации выполняется единичными КАМАК-циклами подпрограммой CSSA, причем на каждое обращение к CSSA тратится около 2ØØ мксек.

На практике часто удобнее пользоваться не стандартным пакетом непосредственно, а подпрограммами, написанными для конкретных модулей аппаратуры КАМАК. Применение таких подпрограмм позволяет оперировать не командами КАМАК, а реальными величинами, так, например, подпрограммы для модулей ЦАП и АЦП используют значения параметров в Вольтах, подпрограммы для цветного дисплея в стандарте КАМАК используют понятия "точка", "вектор", "знак". Обычно такие подпрограммы бывают написаны на Фортране, что позволяет их легко модифицировать при изменениях в аппаратуре самим пользователям и инженерам- разработчикам. Однако, в тех случаях, когда требуется высокая эффективность, эти подпрограммы переписывают на MACRO-11 или PL-11.

В ИЯФ также получили распространение программы - тесты устройств КАМАК, значительно облегчающие проверку, а зачастую и настройку аппаратуры. Централизованная подготовка и сопровождение такого программного обеспечения позволяет сократить сроки внедрения аппаратуры, облегчить ее использование.

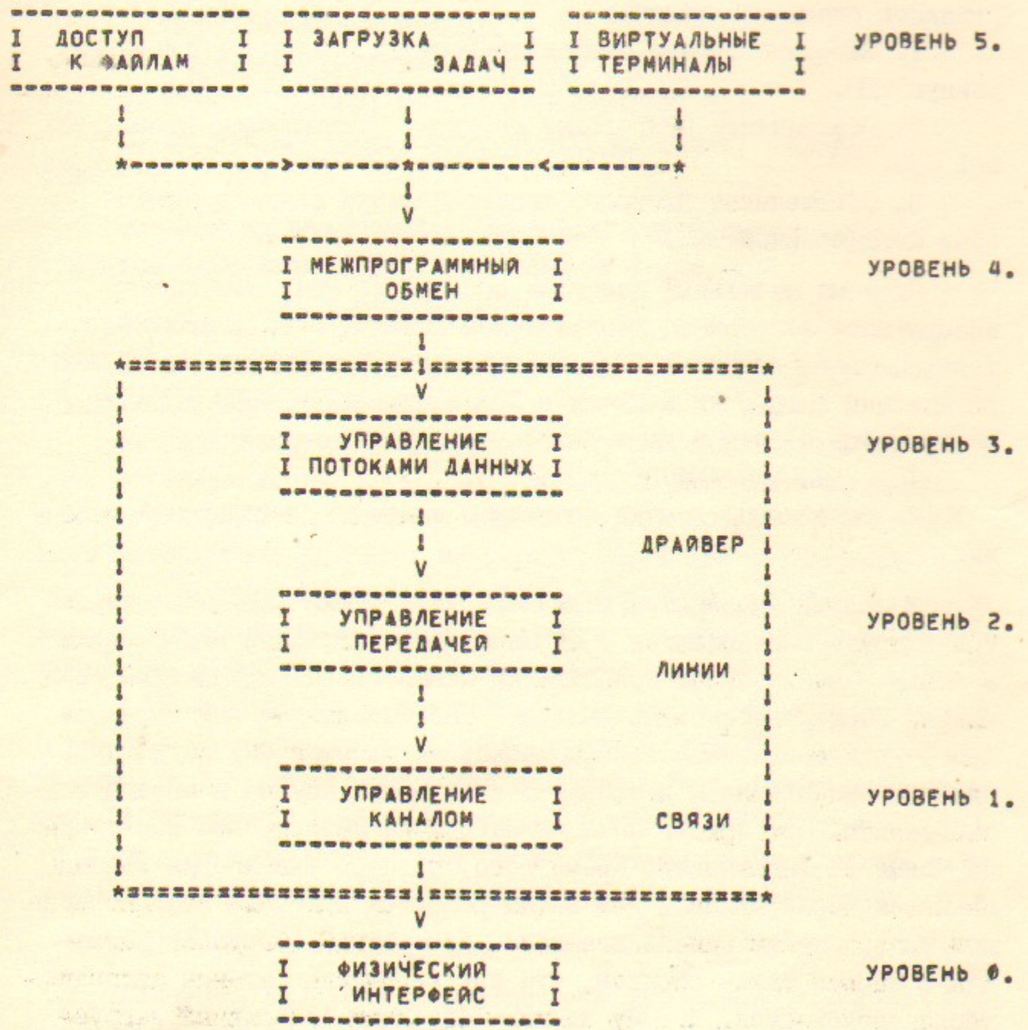

РИС. 1
АРХИТЕКТУРА МАТОБЕСПЕЧЕНИЯ КОМПЛЕКСА

Г Л А В А 3

РЕАЛИЗАЦИЯ СИСТЕМЫ

3.I. Система начальной загрузки.

Система начальной загрузки создавалась как первый этап системы связи, поэтому она обеспечивает возможность самостоятельной работы с терминала периферийной ЭВМ. При ее создании решались следующие задачи:

I. Загрузка образа памяти автономной программы в периферийную ЭВМ.

2. Обеспечение подготовки программ с терминала периферийной ЭВМ.

3. Обеспечение быстрого восстановления связи в случае сбоя периферийной ЭВМ.

Система начальной загрузки состоит из трех компонент: аппаратного загрузчика периферийной машины, модифицированного терминального драйвера и программы загрузки в центральной ЭВМ. Аппаратный загрузчик записан в ПЗУ периферийной ЭВМ и имеет возможность работы в двух режимах:

I. Режим удаленного терминала.

2. Выполнение команд программы загрузки центральной машины.

В режиме удаленного терминала аппаратный загрузчик просто транслирует все принятые с терминала периферийной ЭВМ символы в линию связи, а все принятое из линии связи передает на терминал. Поскольку система RSXIIM не использует бит четности при передаче символьной информации, то восьмой бит каждого символа используется в качестве индикатора команд аппаратного загрузчика. При приеме команды загрузчик меняет свое состояние на время ее выполнения. Кроме того, процесс выполнения команды является неразрывным и тем самым решается проблема информационной прозрачности канала передачи. Аппаратный загрузчик размещен в памяти таким образом, что при включении питания автоматически запускается. К тому же перезапустить аппаратный загрузчик из программы в центральной ЭВМ можно, используя аппаратную возможность интерфейса связи.

3.I.I. Протокол связи.

Протокол связи между аппаратным загрузчиком и программой загрузки BOT в центральной ЭВМ является асимметричным, то есть аппаратный загрузчик полностью управляется из программы загрузки в центральной ЭВМ. Протокол обмена между аппаратным загрузчиком и программой BOT в центральной ЭВМ, имеет следующую структуру: байт команды, один или несколько байтов параметров и байт контрольной схемы. Команды протокола описаны ниже:

I. Передача адреса загрузки.

2. Передача блока данных.

3. Передача стартового адреса.

4. Запрос размера памяти периферийной ЭВМ.

5. Запрос на прием блока данных из памяти периферийной ЭВМ.

Ответом на все команды, за исключением двух последних, является байт вычисленной контрольной суммы.

3.I.2. Аппаратный загрузчик.

Последовательность работы аппаратного загрузчика можно представить следующим образом;

I. Определяется конфигурация и работоспособность аппаратуры и устанавливается связь с центральной ЭВМ. Если загрузчик не обнаруживает консольного терминала, он имитирует его. Это дает возможность работать на периферийной ЭВМ без терминала.

2. Происходит переход в режим удаленного терминала, когда каждый принятый из линии символ с нулевым восьмым битом передается на терминал, а все, что принято с терминала – в линию.

3. Если принята и распознана команда, загрузчик переходит в режим выполнения команд. После принятия команды и до завершения ее выполнения аппаратный загрузчик не обслуживает терминал. Максимальное время, в течение которого он может находиться в этом режиме, составляет около IØØ миллисекунд, после чего загрузчик снова выходит на режим удаленного терминала.

3.I.3. Программа загрузки в центральной ЭВМ (вот).

Программа загрузки в центральной ЭВМ является ведущей при проведении процедуры загрузки. Она выполняет следующие функции:

I. Принимает командную строку с терминала периферийной ЭВМ.

2. Открывает файл с загружаемой программой.

3. Проверяет, что она является автономной программой.

4. Запрашивает размер памяти в периферийной ЭВМ.

5. Передает образ памяти в периферийную ЭВМ.

6. Запускает,если необходимо, загруженную программу.

Программа BOT позволяет также выполнять передачу образа памяти периферийной ЭВМ в файл на дисках центральной ЭВМ.

3.2. Драйвер линии связи.

Драйвер линии связи является основным элементом матобеспечения системы связи, на которой строится межпрограммный обмен и виртуальные устройства. Как уже говорилось выше, он включает три стандартных сетевых уровня протокола обмена, что позволяет весьма существенно экономить память в периферийной ЭВМ, а матобеспечение тех же трех уровней в системе занимает около 6К слов.

3.2.I. Разделение ресурсов.

Для обеспечения многопрограммной работы в периферийной ЭВМ необходимо разделять линию связи между несколькими парами обменивающихся между собой программ. Наиболее распространенным способом такого разделения является введение концепции логического канала. Логический канал принадлежит одной паре программ на все время обмена и работает независимо от других логических каналов с точки зрения этой пары. Логические каналы могут быть фиксированными или создаваться динамически. В системе связи реализованы фиксированные логические каналы, число которых задается во время генерации драйвера и не может быть больше восьми. Они представлены в операционной системе как несколько устройств одного типа,управляемых, для данной линии, одним контроллером.

3.2.2. Синхронизация процессов.

Обмен данными на уровне программ пользователя, работающих асинхронно, требует введения средств синхронизации обмена. В качестве таких средств в системе RSXII используются флаги событий и асинхронные программные прерывания (AST). Для вклю-

чения механизма синхронизации перед началом обмена каждая программа должна захватить логический канал, выставляя запрос на прикрепление канала. Параметрами такого запроса могут быть номер локального флага программы и (или) адрес подпрограммы обработки AST . После того, как программа "А" монополизировала канал, она имеет возможность узнать о состоянии канала на противоположном конце. Если канал на противоположном конце будет монополизирован программой-партнером, для программы "А" будет организовано AST с передачей специального кода, говорящего о том, что можно начинать сессию обмена. То же произойдет, если одна из программ освободит канал. Для синхронизации на уровне передачи блока используются локальные флаги. Например: если программа "А" выдала запрос на чтение блока, то на противоположном конце для программы "В" будет уставлен флаг, заказанный при выполнении запроса на прикрепление канала. Таким образом, обмен будет синхронным, если программа "В" перед выдачей запроса на передачу будет ожидать флага синхронизации. В том случае, если одной из программ, например "А", необходимо изменить режим обмена, она может выдать запрос на передачу синхронизующейся информации пользователя. При этом на противоположном конце в программе "В" произойдет AST так же, как и при прикреплении канала, но переданный код будет тем, который был указан при выдаче запроса программой "А". Всего может быть передано 254 различных кода.

3.2.3. Протокол обмена.

При создании протокола обмена для драйвера линии связи учитывались следующие основные требования:

I. Минимизация памяти, требуемой для оформления сеанса связи.

2. Минимизация времени реакции на проведение сеанса связи.

3. Совместимость с системой начальной загрузки.

4. Возможность быстрого восстановления в случае сбоя.

Наиболее подходящим типом протокола, удовлетворяющим этим требованиям, является интерактивный протокол обмена, то есть такой протокол, в котором диалог ведется перед началом любого сеанса связи и проведение сеанса возможно только в том случае, если оба абонента готовы к сеансу. В противном случае сеанс прекращается с указанием ошибки. Передача ведется порциями ин-

формации, несущими определенную смысловую нагрузку. Такая порция называется кадром. Прозрачность канала передачи информации обеспечивается за счет введения специального символа, который является управляющим и при приеме изменяет состояние драйвера. Этот символ является заголовком служебного кадра. Следующий за ним символ несет служебную информацию. Размер служебного кадра фиксирован и составляет 2 слова. Типы служебных кадров приведены ниже:

1. Терминатор - вызывает прекращение текущего сеанса связи и отключение драйвера от линии связи в центральной ЭВМ.

2. Данные - говорит о том, что управляющий символ необходимо интерпретировать как данные и передать в буфер пользователя.

3. Ответ - несет в себе информацию о том, как завершился прием драйвером на противоположном конце или о возможности начать передачу.

4. Запрос на передачу - говорит о том, что драйвер готов к передаче информации по данному логическому каналу. Содержит информацию о длине передачи.

5. Синхросигнал - требование уставить флаг синхронизации для задачи, прикрепившей канал.

6. Прикрепление - говорит о том, что на противоположном конце логический канал прикреплен с использованием механизма синхронизации.

7. Открепление - говорит о прекращении режима синхронизации на данном логическом канале.

8. Синхронизирующее сообщение пользователя - говорит о том, что необходимо прервать программу, прикрепившую канал и передать ей принятое сообщение (1 байт).

8. Повторитель - требование повторить запрос на передачу, используется для разрешения конфликтных ситуаций.

3.2.4. Обеспечение доступа к файлам центральной ЭВМ.

Для обеспечения возможности работы из периферийной ЭВМ с файлами центральной ЭВМ создан комплекс программных средств: специальный пакет подпрограмм в периферийной машине и программа RFP - процессор удаленных файлов в центральной ЭВМ.

Подпрограммы пакета, в соответствии с поддерживаемым протоколом обмена, передают программе RFP в центральной ма-

шине запросы на выполнение файловых операций открытия, закрытия, чтения и записи. Программа RFP адресует принятые запросы файловой системе центральной ЭВМ для исполнения. В зависимости от типа запроса программа RFP и задача в периферийной ЭВМ обмениваются массивами данных, являющимися либо исходными данными, либо результатом выполнения файловых операций.

В составе пакета, работающего с файлами центральной ЭВМ входят подпрограммы, предназначенные для использования в программах, написанных на языке ассемблера MACRO-11. В рамках пакета существует интерфейс между этими подпрограммами и Фортраном. Для всех подпрограмм входным параметром является так называемый блок описания удаленного файла - RFDB. Этот блок имеет тот же смысл и структуру, что и FDB - блок описания файла в центральной ЭВМ. Задание атрибутов файла производится установкой определенных значений в ячейках RFDB . Это можно сделать с помощью вызова существующих в системе макроопределений. Ошибки выполнения операций над файлами центральной ЭВМ фиксируются в предназначенной для этого ячейке F.ERR .

Запросы из периферийной машины на выполнение операций над файлами принимает программа RFP . Выполнение любого запроса начинается с приема служебного байта - кода запроса. Он несет информацию о требующейся файловой операции и номере логического канала линии связи, по которому будет отрабатываться запрос. Запрос на открытие файла отличается от всех остальных тем, что для него служебный байт передается по нулевому логическому каналу, который всегда прикрепляется программой RFP и готов к приему служебной информации.

При выполнении запросов на открытие и закрытие файлов протокол обмена между процессором удаленных файлов и задачей в периферийной ЭВМ выглядит следующим образом (буква "Ц" символизирует центральную ЭВМ, "П" - периферийную):

Ц. Программа RFP ,принимает код запроса на открытие или закрытие файла. После этого она становится на прием из линии блока описания удаленного файла, по указанному в служебном байте логическому каналу.

2П. Задача в периферийной ЭВМ передает в линию блок описания удаленного файла.

2Ц. Процессор удаленных файлов выполняет заказанную файловую операцию и возвращает в линию блок описания удаленного файла.

Протокол обмена при выполнении файловых операций чтения и записи выглядит иначе. Каждый запрос сопровождается передачей "Блока запроса" - трехсловной структуры, содержащей размер передачи (I слово) и номер виртуального блока в файле, начиная с которого будет производиться чтение или запись данных (2 слова). Ниже следует описание протокола обмена при выполнении файловых операций чтения и записи:

IП. Задача в периферийной ЭВМ передает в центр байт с кодом запроса.

IЦ. После успешного анализа кода запроса программа передает разрешение в периферийную ЭВМ на посылку блока с содержанием запроса.

2П. Получив разрешение, задача в периферийной машине передает процессору удаленных файлов блок с содержанием запроса.

2Ц. Процессор удаленных файлов анализирует правильность составляющих блока запроса и посылает ответ о их корректности.

3П. Задача в периферийной ЭВМ ожидает прихода из линии массива данных (запрос на чтение). Задача в периферийной ЭВМ передает массив данных в линию (запрос на запись).

3Ц. Программа RFP передает в периферийную машину либо прочитанные из файла данные, либо код ошибки выполнения файловой операции (запрос на чтение). Программа RFP принимает из линии блок данных и заносит его в файл. В случае ошибки записи в файл, в линию отдается код ошибки (запрос на запись).

3.3. Доступ к терминалам центральной ЭВМ.

После завершения процедуры начальной загрузки терминальный драйвер отключается от линии связи.

Отсюда следует, что в матобеспечение системы связи должны

быть включены программные средства, позволяющие терминалу периферийной ЭВМ быть терминалом центральной ЭВМ. Для обеспечения такой возможности в операционную систему центральной машины вводится новый тип устройств - виртуальные терминалы. Эти устройства имеют тот же интерфейс с программами и с системой, что и обычные терминалы. Отличие состоит в том, что все принятые запросы передаются в периферийную ЭВМ, где и исполняются терминальным драйвером. Для обеспечения трансляции запроса вводятся два программных процессора, которые выполняют обмен по внутреннему протоколу, используя для передачи линию связи. Таким образом, матобеспечение виртуальных терминалов состоит из трех основных компонент:

I. Драйвера виртуальных терминалов - VT
2. Процессора в центральной ЭВМ - VTP
3. Процессора в периферийной ЭВМ - RMT

3.3.I. Драйвер виртуальных терминалов.

Драйвер виртуальных терминалов является интерфейсом между программами и процессором виртуальных терминалов в центральной ЭВМ. Он принимает запрос на ввод/вывод от программы и передает его процессору VTP . Процессор VTP после исполнения запроса, возвращает управление драйверу, который завершает обслуживание запроса, сообщая о результатах программе.

3.3.2. Процессоры виртуальных терминалов.

Процессоры виртуальных терминалов - это привилегированные задачи, на которые возложены функции трансляции и собственно выполнения запросов. Протокол обмена между ними определяется, во-первых, набором терминальных запросов и, во-вторых, возможностями, которые предоставляет драйвер линии связи. При выполнении запроса программные процессоры обмениваются следующими сообщениями:

I. Сообщение типа "ЗАПРОС" - содержит запрос виртуального драйвера периферийному процессору.

2. Сообщение типа "ОТВЕТ" - содержит ответ после исполнения терминальной функции.

3. Сообщение типа "СТРОКА ДЛЯ MCR " - содержит командную строчку для MCR центральной ЭВМ.

Передача сообщения проводится в два этапа: посылки синхронизирующего сигнала, по которому программа-получатель выставляет функцию на прием, и собственно передачи. Сообщение имеет обязательное поле заголовка и необязательное поле терминальных данных. Заголовок содержит тип сообщения и параметры запроса или статус В/В. Поле терминальных данных имеет переменную длину, которая определяется из заголовка.

Последовательность выполнения запросов в программных процессорах закладывается в таблицу состояний. Каждый тип запроса описывается своей таблицей состояний. Такой подход позволяет исключить многие проверки и, следовательно, упростить внутреннюю логику работы программ.

Центральный программный процессор VTP может поддерживать до восьми удаленных терминалов. VTP запускается однажды программой загрузки системы LDS . Эта же программа передает информацию о номере линии, через которую проводилась загрузка. Процессор VTP подключается к линии и, если периферийный процессор RMT запущен, терминал Э-6Ø представляется как терминал центральной машины, Процессор RMT после старта подменяет МСК и тем самым, решается проблема приема командных строк.

3.3.3. Загрузка нерезидентных задач.

Описываемая система связи обеспечивает динамическую загрузку в периферийную ЭВМ задач, находящихся на дисках центральной машины. Реализована возможность выгрузки задачи из периферийной ЭВМ в файл образа задачи.

Загрузка нерезидентных задач в периферийную ЭВМ осуществляется специальной задачей-загрузчиком. Это стандартная системная программа, измененная для работы в периферийную ЭВМ с описываемой системой связи. Изменения сводятся к замене дисковых операций подпрограммами работы с линией связи. Роль партнера этих подпрограмм в центральной ЭВМ выполняет задача RFP - процессор удаленных файлов. Для обработки запросов на загрузку и выгрузку задач в программе RFP введен специальный режим, который подразумевает работу с дисками помимо файловой системы центральной ЭВМ. Этот режим обеспечивает максимально возможную скорость загрузки.

Задача-загрузчик передает в линию связи запросы двух типов:

I. Запрос на загрузку в периферийную ЭВМ образа задачи, находящегося на диске по указанному физическому адресу.

2. Запрос на выгрузку на диск по указанному физическому адресу определенной области памяти периферийной ЭВМ.

Для работы загрузчика в периферийной ЭВМ выделен отдельный логический канал. На время выполнения каждого запроса загрузчик закрепляет этот канал за собой. Программа ожидает запросов от загрузчика по тому же каналу обслуживаемой линии связи. Протокол обмена между задачей-загрузчиком периферийной ЭВМ и программой RFP центральной машины выглядит следующим образом:

IП. Загрузчик посылает в центр служебный байт с кодом запроса - номером логического канала и типом запроса (загрузка или выгрузка).

IЦ. После успешного анализа кода запроса программа передает разрешение на отправку запроса - трехсловного блока с размером передачи и адресом образа задачи на диске.

2П. Получив разрешение, загрузчик передает в центр блок запроса.

2Ц. Программа RFP анализирует параметры запроса.

3П. Загрузчик ожидает из линии массив данных (запрос на загрузку). Загрузчик передает в линию массив данных (запрос на выгрузку).

3Ц. Программа RFP передает в периферийную машину прочитанные с диска данные блоками определенной длины (запрос на загрузку). Программа RFP принимает из линии данные блоками определенной длины и записывает их на диск (запрос на выгрузку).

3.4. Модификации OTS Фортрана.

Каждая загруженная в память ЭВМ программа содержит в себе часть, взаимодействующую с операционной системой - OTS (OBJECT TIME SYSTEM). OTS Фортрана включает в себя подпрограммы, связанные с файловой системой, обеспечивающие

ввод-вывод, систему диагностики ошибок, арифметические подпрограммы. Наиболее громоздкая часть OTS , взаимодействующая с файловой системой, в RSXIIS не используется, однако, занимает от 5 до 7 К слов. Модификация, исключившая эту часть OTS из образа задачи, позволяет загружать в Электронику-60 программы, написанные на Фортране, длиной более I000 строк.

Модифицированная OTS выполняет следующие функции:

I. Подготовка структур данных OTS при начальном запуске программы.

2. Выдачу на терминал сообщения при возникновении ошибки во время исполнения программы. Такое сообщение содержит номер ошибки, строку и название подпрограммы, где произошла ошибка, всю последовательность вызовов подпрограммы.

3. Подпрограммы для поддержки операторов STOP и PAUSE.

4. Обеспечение принципа умолчания фактических параметров при вызове подпрограмм (с стандартном Фортране это возможно только в отношении подпрограмм, написанных на ассемблере).

Обеспечена совместимость с отладчиком FDT.

3.5. Внутренняя структура и параметры пакета КАМАК.

Как было описано выше, использование драйвера для работы с аппаратурой КАМАК нецелесообразно - с аппаратурой взаимодействуют непосредственно подпрограммы задачи пользователя. Однако, необходима диспетчеризация доступа к крейту, исключающая взаимное влияние задач, связанное с тем, что они используют общий контроллер. Такая диспетчеризация обеспечивается изменением аппаратного приоритета на время, когда пребывание последовательности операций с КАМАК недопустимо.

Такой подход позволил получить следующие временные параметры для Электроники-60:

Время исполнения единичной команды КАМАК (CSSA)	220 мксек
Время исполнения команды КАМАК при передаче массива данных (CLUBS)	25 мксек
Время реакции на запрос L:	

в однопрограммном режиме	200 мксек
в многопрограммном режиме	I мсек.

Различия во времени реакции на запрос L в однопрограммном и многопрограммном режимах связаны с тем, что в последнем случае запрос должен обрабатываться операционной системой.

ПРОИЗВОДИТЕЛЬНОСТЬ И ХАРАКТЕРИСТИКИ КОМПЛЕКСА

Одним из самых важных элементов, оказывающих влияние на производительность комплекса, является интерфейс межмашинной связи. Поскольку при создании данного комплекса использовался разработанный в ИЯФ интерфейс связи DL-Ki/Si с обменом по программному каналу, скорость обмена сравнительно невысока и составляет около четырех килобайт в секунду. Кроме того, интерфейс с обменом по программному каналу значительно снижает производительность операционной системы в центральной ЭВМ. Это накладывает ограничения на количество пользователей, обслуживаемых центральной машиной. Для рассматриваемой конфигурации число таких пользователей составляет 3-4. При дальнейшем увеличении числа пользователей скорость обмена падает, а следовательно увеличивается время реакции на исполнение запроса. Применение для тех же целей интерфейса с прямым доступом к памяти, создаваемого в данный момент в ИЯФ, позволит увеличить скорость обмена примерно на порядок, а производительность операционной системы повысить в 2-3 раза. Создание централизованного архива в рамках локальной сети, состоящей из многомашинных комплексов, позволит решить проблему хранения информации и распределения вычислительных мощностей.

Необходимо заметить, что подход к построению многомашинного комплекса и его архитектура не зависят от типа используемых ЭВМ и их особенностей. Это позволяет использовать описанную архитектуру при создании матобеспечения для комплексов, построенных на базе мини- и микро-ЭВМ других типов.

Литература

I. А.Н.Алешаев, С.Д.Белов, Б.В.Левичев, И.Я.Протопопов. Операционная система ЭВМ ОДРА для управления электрофизическими установками в ИЯФ СО АН СССР. Препринт ИЯФ 80-194, Новосибирск, 1980.

2. В.А.Сидоров, Б.Л.Сысолетин, Б.Н.Шувалов. Программное обеспечение системы "РАДИУС". - "Управляющие системы и машины", 1978, № 1.

3. Б.Н.Шувалов. Программное обеспечение периферийных ЭВМ М-6000 многомашинного комплекса "РАДИУС" - Препринт ИЯФ, 1982.

4. Э.А.Якубайтис. Архитектура вычислительных сетей - Институт Электроники и Вычислительной Техники, г.Рига, 1981.

5. Дж.Мартин. Системный анализ передачи данных - Мир, Москва, 1975.

6. GUIDE TO WRITING AN I/O DRIVERS - DIGITAL EQUIPMENT CORPORATION, MAINARD, MASS., USA, 1977.

7. RSXIIM OPERATORS PROCEDURES MANUAL - DEC, MAINARD, MASS., U S A, 1977.

8. RSXIIM/S I/O DRIVERS REFERENCE MANUAL - DEC, MAINARD, MASS., USA, 1977.

9. "Subroutines for CAMAC" ESONE/SR/01

10. Вьюшин О.В., Храпкин П.Л. "Пакет отандартных подпрограмм для работы с КАМАК". "Автометрия", 1982 г., № 4.

11. IAS/RSXIIM I/O-OPERATIONS REFERENCE MANUAL - DEC, MAINARD, MASS., USA, 1977.

12. С.Д.Белов. Система компиляции на управляющих машинах комплекса ВЭШ-4. Работы молодых специалистов, выполненные в ИЯФ СО АН СССР в 1977-1978 годах. Отчет, Новосибирск, 1978.

М.В.Бейлин, О.В.Вькшин , А.Д.Клименко,
В.В.Конюхов, В.Р.Кузнецов, А.В.Проскурин,
П.Л.Храпкин, А.К.Эпштейн

СИСТЕМНОЕ МАТОБЕСПЕЧЕНИЕ МНОГОМАШИННОГО КОМПЛЕКСА АВТОМАТИЗАЦИИ ФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА

Препринт
№ 82- 72

Работа поступила 28 мая 1982 г.

Ответственный за выпуск С.Г.Попов
Подписано к печати 04.06.82 МН 03359
Формат бумаги 60x90 1/16 Усл.1,9 печ.л., 1,5 уч.-изд.л.
Тираж 290 экз. Заказ № 72 Бесплатно

Ротапринт ИЯФ СО АН СССР, г.Новосибирск-90