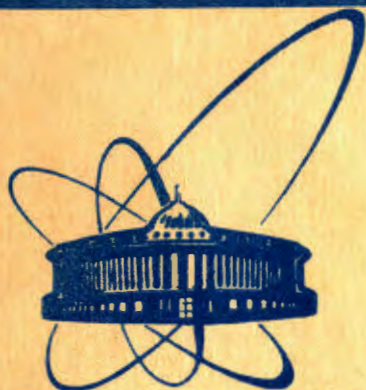


84-90



**сообщения  
объединенного  
института  
ядерных  
исследований  
Дубна**

Ц 841e

Г-53

2480/84

10-84-90

Э.М.Глейбман, В.Е.Жучко, Т.В.Рукояткина,  
В.В.Тарасов

**СТЕНД ДЛЯ СОЗДАНИЯ И РАЗВИТИЯ  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
МИКРОПРОЦЕССОРНЫХ СИСТЕМ**

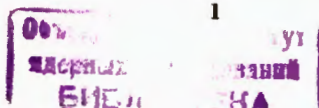
**1984**

Опыт использования микропроцессорной электронной техники в системах автоматизации физических исследований и для управления сложными физическими установками показывает, что кроме общего снижения затрат на создание электронной аппаратуры, значительно улучшатся ее технико-экономические показатели, непосредственно связанные с повышением эффективности физических исследований.

Качество систем, выполненных на базе микропроцессоров и микро-ЭВМ, а также возможности их использования для управления, обработки и передачи данных физического эксперимента, во многом определяются оснащенностью средствами автоматизации разработки программ. Имеющиеся в настоящее время кросс-средства и системы развития программ на прототипных микро-ЭВМ /1,2/ из-за своей сложности или высокой стоимости не всегда доступны разработчикам.

В то же время используемые в физических лабораториях микропроцессорные КАМАК-контроллеры, блоки памяти и интерфейсные модули к стандартной периферии ЭВМ обеспечивают возможности для создания на их основе дешевых и доступных средств автоматизации разработки программ.

Для этих целей в ЛЯП ОИЯИ была создана программа редактора текста и ассемблера для микропроцессоров типа КР580ИК80 /3/. При своей простоте данная программа эффективна лишь при разработке небольших программ из-за ограничений на синтаксис используемого языка ассемблер по сравнению со стандартным /4/, и ограниченных возможностей в редактировании символьных файлов.





В настоящей работе описываются аппаратные и программные средства стенда, выполненного в стандарте КАМАК на основе микро-процессорного контроллера "МИКАМ-2" /5/ и резидентного программного обеспечения.

Стенд предназначен для создания и разработки программ микро-процессорных систем как в стандарте КАМАК, так и для любых систем автоматизации и обработки данных, использующих микропроцессор типа КР580ИК80 (ИНТЕЛ-8080).

Аппаратные и программные средства стенда дают возможность осуществлять:

- подготовку и редактирование исходных программ;
- трансляцию программ, написанных на ассемблере ИНТЕЛ-8080, в объектные модули с выдачей листинга;
- загрузку, отладку и моделирование объектных модулей;
- вывод отлаженных программ на перфоленту, либо запись в ПЗУ, а также другие сервисные функции.

Разработанные для стенда текстовый редактор и ассемблер не имеют ограничений, указанных в /3/, что позволяет создавать программы достаточно больших объемов.

Состав аппаратуры стенда. Блок-схема стенда приведена на рис.1. Кроме микропроцессорного контроллера "МИКАМ-2" и интерфейсных модулей /6,7/ к стандартной периферии, стенд включает модули оперативной памяти объемом 4, 16 и 32 Кбайт (статического и динамического типов), модуль постоянной памяти объемом 8 Кбайт, в котором хранится резидентное программное обеспечение, программаторы ПЗУ.

В зависимости от потребностей конкретных работ модули памяти могут наращиваться по 4 или 16 Кбайт до общего объема - 64 Кбайт. Управление работой стенда производится с одного из терминальных устройств (дисплея или строкочечатающего устройства с клавиатурой

типа DZM-180/KSR ), подключаемых по последовательному каналу связи к контроллеру "МИКАМ-2". Скорость передачи данных между контроллером и терминальными устройствами - 9600 бод.

Для ввода символьных и бинарных лент используется фотосчитывающее устройство FS -150I. Вывод информации для долговременного хранения производится с помощью перфоратора ПЛ-150 (ПЛ-80) /6,7/.

Модуль ПЗУ PROM-8 К /5/ предназначен для хранения резидентного программного обеспечения стенда (программа транслятора, редактора, тесты ОЗУ и т.п.).

Программаторы ПП-8 /8/ и ПП-75 /9/ построены для программирования конкретных типов ПЗУ:

Тип ПЗУ	Организация	Тип программатора
SN74SI88; MN74I88	32x8	ПП-75
SN74S287; SN74S387	256x4	ПП-75
SN74S470; SN74S471	256x8	ПП-75
SN74S472; SN74S473	512x8	ПП-75
SN74S474; SN74S475	512x8	ПП-75
I 2704; K573PFI	1024x4	ПП-75 (ПП-8)
I 2708; K573PFI	1024x8	ПП-75 (ПП-8)

Модуль ПП-78 /9/ представляет собой общую часть логической схемы, не зависящую от типа ПЗУ, к которой при помощи разъема подключается модуль КАМАК со схемой управления и источниками напряжения для конкретного типа ПЗУ. Это позволяет осуществлять быструю разработку программирующих устройств для новых типов ПЗУ. Подключение того или иного программатора к стенду определяется типом используемого ПЗУ.



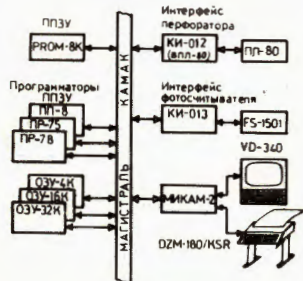


Рис.1. Блок-схема стенда.

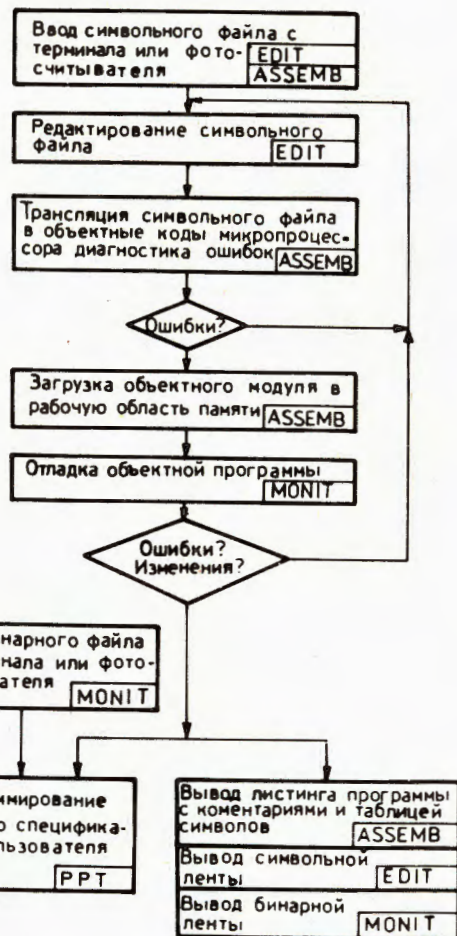


Рис.2. Взаимодействие программных компонент стенда при разработке новых программ.

Программное обеспечение стенда состоит из двух частей. Первая часть – это расширенная мониторная система "МИКАМ" (MONIT), пакет прикладных программ (ППП), текстовый редактор (EDIT) и ассемблер (ASSEMB).

Вторая часть программного обеспечения стенда больше выполняет сервисные функции, она включает:

систему программирования ПЗУ (PPT); программы тестирования ОЗУ (TEST) и реассемблер (DESASS).

Расширенная мониторная система (MONIT) занимает объем внутреннего ПЗУ контроллера около 2 Кбайт и кроме процедур, описанных в /IO/, включает:

- драйверы стандартной периферии;
- дополнительные приказы монитора, используемые для управления периферией, инициализации и вызова резидентных программ (EDIT, ASSEMB, TEST и т.п.).

Оставшиеся 2 Кбайта внутреннего ПЗУ контроллера занимает пакет прикладных программ (ППП). В ППП входит усовершенствованный вариант пакета программ с плавающей запятой /II/, дополненный подпрограммами преобразования кодов и форматов, упаковки и распаковки чисел, а также подпрограммы вычисления алгебраических (квадратный корень, натуральный логарифм, экспонента) и тригонометрических функций. Операции производятся с данными, находящимися во внутренних регистрах микропроцессора, диапазон изменения чисел с плавающей запятой  $2^{-64} + 2^{63}$ .

Кроме того, в ППП включено около двадцати процедур, наиболее часто встречающихся при работе с модулями КАМАК, процедуры обработки цифровой и текстовой информации, преобразование форматов.

Порядок разработки программного обеспечения в системе МИКАМ представлен на рис.2.

Исходный символический текст программы на ассемблере вводится в память с терминала или фотосчитывателя и обрабатывается программой-редактором (EDIT). Редактор позволяет вносить изменения в текст: вычеркивать и вставлять символы и строки, сшивать файлы. На следующем этапе отредактированный символический файл транслируется в объектные коды микропроцессора и сразу же загружается в рабочую область памяти программой-ассемблером (ASSEMB). Ассемблер осу-



цествляет диагностику ошибок в обозначениях регистров, команд и операндов, в определении меток, сигнализирует о переполнении таблицы символов. Ориентируясь на сообщения об ошибках трансляции, разработчик снова может переходить к программе - EDIT, внося изменения в текст и добиваясь правильности трансляции и выполнения программы. Такая возможность отладки программы от начала и до конца обеспечивается одновременным хранением в памяти как объектного, так и символьного файла в рабочей области редактора. После каждого запуска программы и обнаружения ошибки в текст оперативно вносятся изменения с помощью редактора, а ассемблер корректирует абсолютные шестнадцатичные адреса объектного файла. Окончательная доводка программы осуществляется непосредственно на аппаратуре стенда с использованием отладочных возможностей монитора.

Такая работа особенно удобна и полезна, когда производится разработка программы для систем КАМАК. В этом случае отладке и проверке могут быть подвергнуты не только вычислительные процедуры, но и программы, непосредственно связанные с управлением аппаратурой КАМАК. Результатом такой работы могут явиться дополнения и изменения в исходный файл, которые могут быть тут же внесены с помощью программы EDIT.

Когда объектный модуль полностью отлажен и в редактировании больше нет необходимости, по желанию оператора он может быть выведен на перфоленту, либо записан в один из типов ПЗУ с помощью программы PPT.

С помощью программ EDIT и ASSEMB могут быть получены символьные ленты объектных модулей, а также листинги текста программы с комментариями и полные или частичные таблицы символов ассемблера с учетом всех изменений.

Объем разрабатываемых программ определяется объемом доступной оперативной памяти контроллера (64 Кбайта). Обычно под таблицу сим-

волов отводится память 4 Кбайта и более. Оставшееся адресное поле за вычетом объема ПЗУ и внутренней памяти контроллера распределяется между объектным файлом (рабочей программой) и символьным файлом (рабочей областью редактора).

Редактор текстов (EDIT) является строчноориентированным. Приказы редактора позволяют исправлять целые строки или часть строки, вставлять в любое место текста символы, строку или несколько строк, осуществлять поиск нужной строки по образцу и осуществлять замену этого образца на другой. Длина образцов текста в приказах не превышает 70 знаков. Максимальная длина редактируемого текста ограничена размерами используемой памяти. Полный набор приказов редактора с кратким описанием выполняемых действий приводится ниже в таблице.

Таблица приказов редактора текстов

SP - пробел, LF - перевод строки, \* - номер редактируемой строки,  
CR - возврат каретки, N - десятичное число,

№ пп	Приказ	Что исправить или чем заменить	Конец приказа	Расшифровка функций
I	2	3	4	5
1.	N	-	CR	- перейти на строку с номером N
2.	/	N	CR	- перейти на строку с номером * + N
	/	-	CR	- перейти на строку с номером * + I
3.	↑	N	CR	- перейти на строку с номером * - N
	↑		CR	- перейти на строку с номером * - I
4.	L	N	CR	- напечатать N строк, начиная с *
5.	-	N	CR	- стереть N строк
6.	-		CR	- стереть одну строку
7.	SP	Текст	CR	- заменить строку новым текстом



I	2	3	4	5
8.	.	Текст	CR	- вставить после строки * текст
9.	F	Текст1/Текст2	CR	- найти строку, содержащую текст1
10.	G	Текст1/Текст2	CR	- заменить в строке комбинацию Текст1 на комбинацию Текст2
11.	B	Текст1/Текст2	CR	- замена во всех строках, начиная со строки с номером *
12.	C	новый текст	CR	- замена отдельных символов в строке *
13.	U	текст	LF	- вставить текст произвольной длины до строки *
14.	;	текст	LF	- вставить текст произвольной длины после строки *
15.	T	№1, №2, №3	CR	- текст, начиная со строки №1 по строку №2, вставить после строки №3, на старом месте текст стереть
	T	№1, №2, №3	SP	- то же самое, но без стирания исходного текста
16.	D	N	CR	- вставить строку с номером N после строки *
	D		CR	- дублировать строку *
17.	X	A1, A2	CR	- установление границ рабочей области памяти редактора, A1 и A2 в HEX форме
18.	M			- выход в монитор, МИКАМ-2
19.	A			- выход в ассемблер
20.	S			- печать на терминале объема свободной памяти
21.	?			- последовательное дублирование на перфораторе ленты, вводимой с фотосчитывателя
22.	O			- распечатывание текста без нумерации строк
23.	E			- запись в редактируемую строку строки EOF, CR, LF - признака конца файла
24.	R		CR	- ввести с фотосчитывателя текст и поместить в начало области памяти, заданной приказом X
	R		SP	- ввести с фотосчитывателя текст и поместить в память, начиная со строки с номером *

I	2	3	4	5
	R	N	CR	- ввести текст и вставить между строками N и N+1
25.	P		CR	- вывести на перфоленту весь текст из рабочей области памяти
	P		SP	- вывести на перфоленту текст, начиная со строки * до конца
	P	N	CR	- вывести на перфоленту N строк, начиная со строки *
	P	A1, A2	CR	- вывести на перфоленту содержимое памяти между адресами A1 и A2 в HEX формате
26.	\			- перфорация 20 см пустой ленты
27.	@			- перфорация признака конца файла для HEX формата

Редактор занимает в ППЗУ объем, равный 1,5 Кбайт.

Программа-транслятор. Программа ASSEMB осуществляет трансляцию символического файла в объектный с одновременной загрузкой его в рабочую область памяти.

Приказы программы ASSEMB имеют два различных формата, которые обязательно начинаются с точки. Первая группа приказов - это идентификаторы ввода/вывода, вторая группа - приказы управления трансляцией.

Приказы ввода-вывода задают режим пересылки информации на периферийные устройства и с них. Разрешены следующие приказы:

$$\cdot AL = \begin{cases} NO \\ TTY \\ PTR \end{cases} \quad \cdot AP = \begin{cases} NO \\ TTY \\ PTR \end{cases}$$

$$\cdot AR = \begin{cases} NO \\ TTY \\ PTR \end{cases} \quad \cdot AM = \begin{cases} NO \\ RAM \end{cases}$$

где AL - листинг;  
AR - чтение информации;  
AP - вывод информации;  
AM - память;



TTY - терминал;

PTP - перфоратор;

PTR - фотосчитыватель;

RAM - ОЗУ;

NO - устройство отсутствует (отмена устройства).

Управление трансляцией начинается с приказа инициализации ассемблера .I, который резервирует заданную оператором область памяти для таблицы символов и выбирает устройства ввода-вывода и источник транслируемой программы:

.AL = TTY      .AP = PTP

.AR = PTR      .AM = RAM

.F

где .F означает трансляцию программы из памяти (.O - соответственно с перфоленты). Текст транслируемой программы может находиться на перфоленте, либо быть записанным в память с помощью редактора. Обычно текст находится в памяти. Это позволяет транслировать программу, и если есть ошибки, то исправлять их при помощи редактора, не выводя текст на перфоленту, что существенно экономит время отладки программ, а также исключает возможность возникновения новых ошибок, поскольку отпадает необходимость в промежуточном хранении символического или объектного файла на бумажной ленте.

В ходе трансляции и отладки оператор может выбирать периферийные устройства новыми приказами ввода/вывода.

Когда источник транслируемой программы и устройства ввода/вывода выбраны, ассемблер переводится в режим трансляции (of line) приказом .R. Оператор имеет возможность полностью контролировать процесс трансляции, пользуясь сообщениями об ошибках трансляции, а также приказами управления трансляцией. У него есть возможность контролировать запись кодов транслируемой программы в память (приказ .AM), прервать трансляцию нажатием любой клавиши (возврат в режим on line), продолжать ее (.C), распечатать или за-

помнить таблицу символов ассемблера (.H, .W), или ввести новую (.J), вычеркнуть ненужные символы из таблицы (.K), распечатать определенные и неопределенные символы (.S и .U), вывести объектный файл (.P, .B), передать управление какой-либо подпрограмме, редактору для исправления замеченных ошибок или монитору (.G SP, .E, .M).

В результате трансляции получается объектный модуль программы. Одновременно происходит его загрузка в рабочую область памяти, что очень удобно для пользователя, поскольку не требует вывода объектного модуля на ленту и последующей загрузки. Сразу после трансляции можно производить запуск и отладку программы.

Ассемблер позволяет разработчику вывести отлаженный объектный модуль на ленту как в двоичном (.M, W A1 A2), так и в шестнадцатиричном формате (.P или .B), а также получить полный листинг программы (.D, .R) в следующем виде:

1. Номер строки.
2. Абсолютный шестнадцатиричный адрес.
3. Машинный код инструкции.
4. Младший байт операнда.
5. Старший байт операнда.
6. Оригинальный ASCII текст программы с комментариями.

Символьный текст транслируемых программ должен соответствовать требованиям, предъявляемым к синтаксису языка ассемблер INTEL-8080 / 4 / без макрокоманд. Объем, занимаемый программой ASSEMB в ПЗУ, равен 4 Кбайта.

Программное обеспечение программаторов ПЗУ состоит из двух частей. Первая часть для модулей ПР-75 и ПР-78, как более универсальных и предназначенных для программирования многих типов ПЗУ, оформлена как резидентное программное обеспечение и хранится в блоке PROM-8K. Программа управления для модуля ПП-8 хранится на бумажной ленте и вводится в память при необходимости. Обе програм-



мы (объемом около 1 Кбайта) работают в режиме диалога, в процессе которого пользователь простыми приказами с терминала задает режим записи, считывания или контроля записанной в ПЗУ информации. Отличие состоит лишь в том, что при работе с модулями ПР-75 или ПР-78 они должны устанавливаться на фиксированном месте в крейте, в то время как модуль ПР-8 может быть установлен на любое свободное место - в некоторых применениях это предпочтительнее. Для исключения ошибок при программировании ПЗУ из-за неисправности программаторов в каждой программе на этапе инициализации введены процедуры автоматического тестирования модулей с индикацией.

Программа обратной трансляции (DESASS) позволяет получить листинг содержимого области памяти в виде символического ASCII текста с абсолютными адресами, что позволяет разобраться в незнакомой программе. DESASS дает возможность произвести модификацию адресов в программе при перемещении ее в новую область памяти с печатью или без печати модифицируемых адресов, а также найти в программе любой код по заданному образцу. Режим работы программы, а также границы реассемблируемого массива оператор задает в виде простых приказов.

Программа хранится на бумажной ленте и вводится в память при необходимости. Занимаемый объем - около 1,5 Кбайт.

Тест функционального контроля оперативной памяти состоит из двух тестов.

Тест I реализует алгоритм "запись-считывание" и осуществляет последовательную запись и считывание в контролируемую область памяти:

- а) нулей;
- б) единиц;
- в) единиц в нечетные разряды;
- г) единиц в четные разряды;
- д) информацию из ПЗУ монитора контроллера.

Тест 2 является модификацией алгоритма "бегущая 1".

В обоих тестах область контролируемой памяти задается с терминала в виде начального и конечного адресов. Имеется возможность организации циклического выполнения тестов. Число циклов от 1 до 255. Индикация об ошибке выдается на терминал в виде адреса, по которому произошел сбой, считываемой информации и эталона. Тест хранится в постоянной памяти ПЗУ-8К и занимает объем около 0,5 Кбайт.

Описанный стенд для разработки программного обеспечения представляет собой законченный комплекс аппаратных и программных средств, позволяющий производить отладку программы от начала до конца. Одновременно с трансляцией исходного файла происходит его загрузка в рабочую область памяти, что экономит время пользователя, поскольку не требует вывода объектного файла на ленту и последующей загрузки. Большим удобством стенда является одновременное хранение в памяти как рабочего, так и исходного символического файла. Это позволяет оперативно исправлять ошибки трансляции при помощи редактора и добиваться правильности выполнения программы.

Описанный стенд более двух лет используется в Лаборатории ядерных реакций ОИЯИ для развития программного обеспечения систем активационного анализа на микротроне МТ-22, а также в Отделе новых методов ускорения ОИЯИ для развития программного обеспечения технологических систем контроля и настройки электронной аппаратуры управления тяжелоионного синхротрона /12/.

#### Литература

1. Fugman G., Werner D. Cross-Assembler Burden INTEL 8080 TV. Dresden 08-13-76.
2. MDS-800 Intellec. MDS Microcomputer Development System. Santa Clara, 1975.
3. Чурин И.Н. ОИЯИ, IO-I2679, Дубна, 1979.



4. INTEL 8080 Assembly Language Programming Manual. INTEL Corp., 1975.
5. Глейман Э.М., Тарасов В.В. Приборы и системы управления, 1980, № 12, с.16.
6. Антюхов В.А. и др. ОИЯИ, IO-II636, Дубна, 1978.
7. Семенов Б.Ю. и др. ОИЯИ, I3-8I-27I, Дубна, 1981.
8. Глейман Э.М., Каржавин В.Ю. ОИЯИ, P10-80-223, Дубна, 1980.
9. Бамбуров Н.С. и др. Программаторы программируемых постоянных запоминающих устройств. Препринт ИФВЭ, 82-25, ОЭА, Серпухов, 1982.
10. Гласнек К.П., Глейман Э.М. ОИЯИ, P10-I27000, Дубна, 1979.
11. Caserta J. 8080 Floating Point Package with BCD Conversion Routine. Microcomputer User's Library Submittal Form. Ref. No 80-80.
12. Александров В.С. и др. В сб.: "Ускорительный комплекс тяжелых ионов в ОИЯИ". ОИЯИ, P9-83-6I3, Дубна, 1983, с.159-168.

Рукопись поступила в издательский отдел  
14 февраля 1984 года.

#### НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

Вы можете получить по почте перечисленные ниже книги, если они не были заказаны ранее.

	Труды VI Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1978 /2 тома/	7 р. 40 к.
	Труды VII Всесоюзного совещания по ускорителям заряженных частиц, Дубна, 1980 /2 тома/	8 р. 00 к.
D11-80-13	Труды рабочего совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике, Дубна, 1979	3 р. 50 к.
D4-80-271	Труды Международной конференции по проблемам нескольких тел в ядерной физике. Дубна, 1979.	3 р. 00 к.
D4-80-385	Труды Международной школы по структуре ядра. Алушта, 1980.	5 р. 00 к.
D2-81-543	Труды VI Международного совещания по проблемам квантовой теории поля. Алушта, 1981	2 р. 50 к.
D10,11-81-622	Труды Международного совещания по проблемам математического моделирования в ядерно-физических исследованиях. Дубна, 1980	2 р. 50 к.
D1,2-81-728	Труды VI Международного семинара по проблемам физики высоких энергий. Дубна, 1981.	3 р. 60 к.
D17-81-758	Труды II Международного симпозиума по избранным проблемам статистической механики. Дубна, 1981.	5 р. 40 к.
D1,2-82-27	Труды Международного симпозиума по поляризационным явлениям в физике высоких энергий. Дубна, 1981.	3 р. 20 к.
P18-82-117	Труды IV совещания по использованию новых ядерно-физических методов для решения научно-технических и народнохозяйственных задач. Дубна, 1981.	3 р. 80 к.
D2-82-568	Труды совещания по исследованиям в области релятивистской ядерной физики. Дубна, 1982.	1 р. 75 к.
D9-82-664	Труды совещания по коллективным методам ускорения. Дубна, 1982.	3 р. 30 к.
D3,4-82-704	Труды IV Международной школы по нейтронной физике. Дубна, 1982.	5 р. 00 к.
D2,4-83-179	Труды XV Международной школы молодых ученых по физике высоких энергий. Дубна, 1982.	4 р. 80 к.
	Труды УШ Всесоюзного совещания по ускорителям заряженных частиц. Протвино, 1982 /2 тома/	11 р. 40 к.
D11-83-511	Труды совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике. Дубна, 1982.	2 р. 50 к.
D7-83-644	Труды Международной школы-семинара по физике тяжелых ионов. Алушта, 1983.	6 р. 55 к.
D2,13-83-689	Труды рабочего совещания по проблемам излучения и детектирования гравитационных волн. Дубна, 1983.	2 р. 00 к.

Заказы на упомянутые книги могут быть направлены по адресу:  
101000 Москва, Главпочтамт, п/я 79  
Издательский отдел Объединенного института ядерных исследований



**ТЕМАТИЧЕСКИЕ КАТЕГОРИИ ПУБЛИКАЦИЙ  
ОБЪЕДИНЕННОГО ИНСТИТУТА ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ**

Индекс	Тематика
1.	Экспериментальная физика высоких энергий
2.	Теоретическая физика высоких энергий
3.	Экспериментальная нейтронная физика
4.	Теоретическая физика низких энергий
5.	Математика
6.	Ядерная спектроскопия и радиохимия
7.	Физика тяжелых ионов
8.	Криогеника
9.	Ускорители
10.	Автоматизация обработки экспериментальных данных
11.	Вычислительная математика и техника
12.	Химия
13.	Техника физического эксперимента
14.	Исследования твердых тел и жидкостей ядерными методами
15.	Экспериментальная физика ядерных реакций при низких энергиях
16.	Дозиметрия и физика защиты
17.	Теория конденсированного состояния
18.	Использование результатов и методов фундаментальных физических исследований в смежных областях науки и техники
19.	Биофизика

Глейбман Э.М. и др.

10-84-90

Стенд для создания и развития программного обеспечения микропроцессорных систем

На базе микропроцессорного контроллера, стандартных интерфейсных модулей и резидентного программного обеспечения разработан стенд в стандарте КАМАК, с помощью которого производится разработка новых программ для систем автоматизации, использующих микропроцессор типа КР580ИК80.

Аппаратные и программные средства позволяют осуществлять:

- подготовку и редактирование исходных программ;
- трансляцию программ, написанных на ассемблере ИНТЕЛ-8080, в объектные модули с выдачей листинга;
- загрузку, отладку и моделирование объектных модулей;
- вывод отлаженных программ на перфоленту либо запись в ППЗУ, а также другие сервисные функции.

Работа выполнена в Отделе новых методов ускорения ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1984

Перевод авторов.

Gleibman E.M. et al.

10-84-90

The Set-Up for Creating and Development the Microcomputer System's Software

Standard CAMAC Set-up, based on Microcomputer Crate Controller, standard interfaces and resident Software, was developed for program debugging for automatic microcomputer Systems, using КР580УК80. Hardware and Software allows:

- source program introducing and text editing;
- source program translation from INTEL 8080 Assembler to the object module with listing;
- object module loading and debugging;
- complete program output on paper-tape puncher or writing in PROM, end etc.

The investigation has been performed at the Department of New Acceleration Methods, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1984