

Рис. 1. 27697

ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ
им. Г.И. Будкера СО РАН

Т.В. Саликова

СИСТЕМА СБОРА ДАННЫХ УСТАНОВКИ ГДЛ

ИЯФ--92-42

ИЯФ 92-12



НОВОСИБИРСК

ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ им. Г. И. Будкера

Т. В. Салкина

СИСТЕМА СБОРА ДАННЫХ УСТАНОВКИ ГДЛ

ИЯФ-92-42

**НОВОСИБИРСК
1992**

Система сбора данных установки ГДЛ

Т.В. Салжкова.

Институт ядерной физики им. Г.И. Будкера
630090, Новосибирск, Россия

АННОТАЦИЯ

Система сбора данных DAS (Data Acquisition System) работает в среде RSX-11M с аппаратурой КАНАК. DAS является системой, выполняющей операции управления и контроля, производящей сбор данных и архивирование. В состав DAS входят три основные утилиты: 1). EDITAC - редактор буферного банка. (В буферном банке находятся описатели КАНАК модулей и описания алгоритмов их работы, а также хранится собранная и обработанная информация). 2). EXCGDL - выполняет сбор данных. 3). ARCHIV - заносит буферные банки в архив и ведет бортовой журнал архивирования. Для поддержки работы КАНАК аппаратуры, в ядро системы включены: КАНАК драйвер (CCDRV) и CAMAC Ancillary Control Processor (CAMACP), осуществляющий защиту модулей при многозадачном режиме работы с КАНАК крейтами. Используя библиотеку подпрограмм DAS, можно создавать задачи для управления и контроля, для анализа и графического представления собранной информации. На имеющемся оборудовании достигнута скорость обмена между задачей и КАНАК модулем 100 К слова/сек. Время доступа к данным, находящимся в архиве или буферном банке, равно 2 К слова/с. (файлы находятся на винчестере 'ST-225 SeaGate').

© Институт ядерной физики им. Г.И. Будкера СО РАН

1. ОПИСАНИЕ СТРУКТУРЫ DAS

При создании DAS были учтены особенности работы установки "газодинамическая ловушка" - Г Д Л [1]:

1). Объемы данных, полученные за один выстрел (эксперимент), могут достигать 200 К слов, а в течение дня производится от десяти и более выстрелов. Следовательно, требуется база данных большой емкости с минимальным временем доступа.

2). Экспериментаторы в некоторых случаях работают независимо, каждый со своей диагностической аппаратурой. Следовательно, необходимо каждому иметь свою базу данных - буферный банк, где описано его КАМАК оборудование. Это позволит одновременно нескольким экспериментаторам работать со своей аппаратурой, не мешая друг другу. Обычно эксперимент на ГДЛ требует синхронной работы всей аппаратуры, что осуществляется путем объединения частных буферных банков в один общий, и передачей управления аппаратурой ведущему экспериментатору.

3). В буферном банке должны храниться: описания КАМАК аппаратуры, и алгоритмы работы с ней, также информация, прочитанная из КАМАКа (raw data), и предварительно обработанные данные (user data). После выстрела информация, прочитанная из КАМАКа, помещается в буферный банк. Задачи, контролируемые диагностическую аппаратуру, должны прочитать данные из банка, обработать, и если нужно, внести в буферный банк обработанную информацию. Завершающая операция - это внесение буферного банка в архив. На выполнение всех этих операций должно уходить не более трех или четырех минут, что определяется интервалом между выстрелами.

4). Архив, как из кубиков, складывается из буферных банков.

Архив должен вмещать неограниченно большие объемы информации, иметь минимальное время доступа, его структура должна позволять пользовательским задачам читать данные тем же способом из архива, как и из буферного банка. Необходимо иметь возможность вносить буферный банк в архив, и извлекать банк из архива. Удобно иметь несколько архивных файлов, в каждом архивном файле размещаются буферные банки, составляющие единую серию выстрелов. В специальном файле - бортовом журнале, фиксируется какой банк куда помещается, т.е. создается запись: выстрел номер N серии "XXXX" находится в архивном файле "DW1: [5, 1] ARCHIV.DSC; 1".

5). Выбранный механизм хранения собранной информации позволяет вести обработку данных в ходе эксперимента - режим on-line, или после завершения экспериментов - режим offline, одними и теми же анализирующими задачами.

6). Т.к. в наличии были только компьютеры "Электроника MC1212" и интерфейс КАМАК [5], позволяющие подключить по четыре креста, необходимо было создать локальную сеть типа "кольцо" Рис. 1. На ГДЛ для управления, контроля и сбора данных достаточно трех компьютеров, которые связаны между собой двухканальным терминальным интерфейсом И12 (протокол RS-232, скорость 9600 бод).

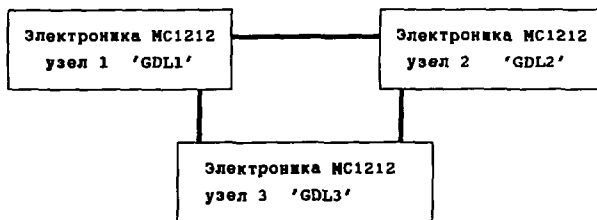


Рис. 1. Локальная сеть ГДЛ типа "кольцо".

На всех компьютерах работает RSX-11M v4.1, а сеть DECnet позволяет вести передачу файлов, работать в режиме удаленного терминала, связывать две задачи в соседних компьютерах [2]. На Рис. 2 показана конфигурация компьютера: процессор М6 (250 тысяч опера-

ций в секунду), четырехканальный терминальный интерфейс DLV-4 с подключенными тремя терминалами (логические адреса: TT0:, TT5:, TT6:) и одним принтером (LP0:). Внешние носители - это два винчестера "ST-225 Seagate" (DW0: и DW1:) емкостью 20 Мбайт, магнитофон (MT0:) CMS300.01. ОЗУ - одна плата памяти 1 Мбайт. Электронный диск (DE:) емкостью 1 Мбайт. Два КАМАК интерфейса с шестью подключенными крейтами (CC0:, CC1:, CC2:, CC3:, CC4:, CC5:); [5]. КАМАК интерфейс и крейт контроллеры соединены оптико-волоконными кабелями. Подключение КАМАК аппаратуры к компьютеру без промежуточных связей устройств сокращает аппаратные и программные расходы при организации сбора данных, т.к. работа с КАМАК модулями осуществляется через стандартный системный механизм - драйвер. Заводской двухканальный интерфейс И12 используется для межкомпьютерной связи. (Более девяти плат в "Электронике ИС1212" не входят).

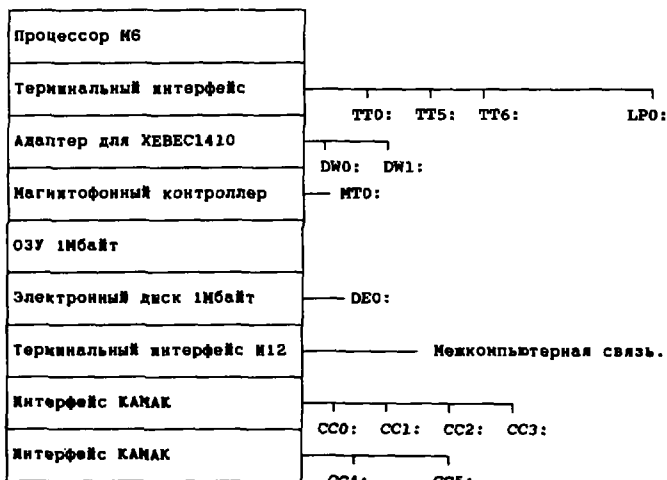


Рис. 2. Конфигурация "Электроника ИС1212".

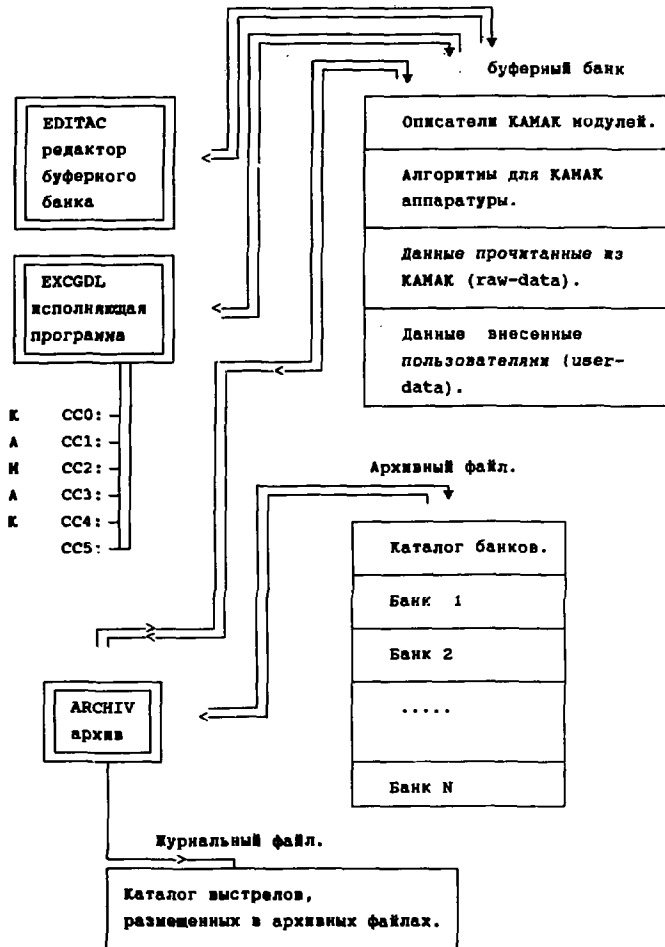


Рис. 3. Логическая схема DAS.

Логическая схема DAS приведена на рис. 3, где указаны обрабатываемые утилитами объекты. Редактор буферного банка EDITAC создает описания алгоритмов работы КАМАК модулей и позволяет изменять параметры их работы. Каждый экспериментатор имеет свой банк. Если необходимо, частный банк можно внести в буферный банк ведущего экспериментатора. EXCGDL - исполняет алгоритмы работы КАМАК оборудования, посылая в крейты NAFы. Данные, прочитанные из КАМАКА (raw-data), вносятся в банк, каждый набор данных имеет свой идентификатор (это имя из шести символов), который используется при чтении данных из банка или архива. По завершению сбора данных EXCGDL запускает задачи, указанные в файле REQUEST.JOB (предназначенные для контроля и анализа работы диагностических систем), и помещает буферный банк в архив. ARCHIV - задача, которая помещает банки в архивный файл и ведет бортовой журнал LOGBOOK.SMT.

1.2. КАМАК-ДРАЙВЕР CCDRV

Для поддержки периферийного оборудования - КАМАК крейтов в среде RSX-11M, написан КАМАК драйвер CCDRV, позволяющий вести обмен данными с модулями крейта в терминах: "чтение логического блока" - в крейт контроллер полагается NAF, и читается из модуля массив слов; или "запись логического блока" - в контроллер полагается NAF, а массив данных заносится в модуль. КАМАК-драйвер может работать максимум с тридцатью шестью крейт-контроллерами типа CC-11 [5]: K16, K606, K607 и т.п. Драйвер обслуживает запросы на чтение и запись данных в модули КАМАК [3]. Запросы завершаются: если происходит прерывание, если модуль выставляет Q, если возникает ошибка при исполнении запроса.

Драйвер выполняет передачу N слов за время T равное:

$$T = T_{qio} + N * t + T_{drv} ,$$

T_{qio} - время исполнения запроса равно примерно 3.7 мс.

t - среднее время исполнение команды равно примерно 5 мкс.

T_{drv} - время, потраченное драйвером для исполнения запроса, может быть от 0.5 до 1.0 мс. (Это 100 - 200 команд).

(Оценки времени проведены для процессора M6). 1 Слово между КАМАК модулем и задачей передается за 9 мс. Для сравнения: аппаратные расходы на передачу одного килобита составляют 4 мс.

1.2.1. Запрос на чтение логического или виртуального блока

Чтение логического блока - операция состоит из исполнения NAPA и чтения из модуля массива данных:

QIOW\$ IO.RLB, LUN, EVENT, PRI, IOSB, AST, <ADR, SIZE, TIM, WORD, , >

Чтение виртуального блока:

QIOW\$ IO.RVB, LUN, EVENT, PRI, IOSB, AST, <ADR, SIZE, TIM, WORD, , >

LUN - логический номер, который назначается на немонтированный крайт. Если крайт монтированный, то LUN назначается модулю.

EVENT - флаг завершения процесса.

PRI - Приоритет запроса.

IOSB - Двухсловный буфер, куда помещается код завершения запроса.

AST - Адрес подпрограммы асинхронного подпроцесса.

ADR - Адрес буфера данных.

SIZE - Размер передаваемых данных в байтах.

TIM - Содержит параметры режима исполнения запроса:

15	14	13	12	08	07	00
LAM	F10	F26	резерв	таймаут запроса		

LAM - Если бит установлен, организуется процесс обработки прерывания. Если установлены биты F10 и F26, то выполняются ниже описанные операции с LAMом.

F10 - Если бит установлен, то при завершении запроса установленный LAM будет сброшен по F10.

F26 - Если бит установлен, то в начале обработки запроса исполняется F26 (разрешение LAM).

Таймаут запроса - время в секундах, отводимое для исполнения запроса. (Если операция не выполняется за этот интервал времени, то запрос завершается с ошибкой).

WORD - Содержит параметры NAF:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Q	F16	F8	F4	F2	F1	N16	N8	N4	N2	N1	A8	A4	A2	A1	W24

Q - Если бит установлен, при завершении запроса происходит проверка Q.

Биты с четырнадцатого по десятый образуют поле кода функции. Биты с девятого по пятый образуют поле номера модуля в крейте.

(В запросе на чтение виртуального блока содержимое этого поля игнорируется).

Биты с четвертого по первый образуют поле субадреса [A8-A1].

W24 - Если бит установлен, идет передача двадцатичетырех-битовых словани между КАМАК-модулем и задачей.

Если бит погашен, передаются шестнадцатибитовые слова.

Пример организации запросов к КАМАК модулю "ЗИИС-4":

1). Организуем запрос, содержащий FO A0, и завершающийся по прерыванию, после того как модуль выполнит измерение. Установим биты: LAM, F26 и F10. Вначале будет исполнено F26 - разрешение LAM. Затем будет исполнен NAF - FO A0. Когда пройдет внешний запуск, модуль выполнит измерение, затем установит LAM, произойдет прерывание. Запрос завершается сбросом LAM по F10. Модуль находится в восьмой позиции, а NAF содержит F10 A0. Читается два байта в буфер "ADDRESS".

```
QIOW$ IO.RLB,LUN,EVENT,IOSB, <ADDRESS,#2,#160405,#024400, >
```

2). Далее следует запрос на чтение в буфер DATA сигнала, снятого с нулевого канала исполняется функция FO A0.

```
QIOW$ IO.RLB,LUN,EVENT,IOSB<DATA,#2,#405,#100400">
```

1.2.2. Запрос на запись логического или виртуального блока

Запись логического блока - запрос на исполнение NAFa и запись массива данных в модуль:

```
QIOW$ IO.WLB,LUN,EVENT,PRI,IOSB,AST,<ADR,SIZE,TIM,WORD, >
```

Запись виртуального блока:

```
QIOW$ IO.WVB,LUN,EVENT,PRI,IOSB,AST,<ADR,SIZE,TIM,WORD, >
```

Функциональное значение параметров на запись не отличается от параметров запроса на чтение.

1.2.3. Запрос на исполнение блока кодов

Специальный запрос на исполнение блока кодов предназначен для подачи нескольких NAF в одном запросе. Содержимое буфера яв-

ляется программой в кодах PDP-11. В блоке кодов может находиться до ста команд, а размер блока не должен превышать 3584 слов. Часть блока кодов используется как буфер данных. Максимальное время, затраченное на исполнение блока кодов не более 500 мкс. Запрос предназначен для неинтерпретированного крейта!

QIOW\$ IO.LOD,LUN,EVENT,PRI,IOSB,AST,<ADR,SIZE,TIM,,,>

LUN - Логический номер, закрепленный за крейтом.

EVENT - флаг завершения запроса.

PRI - Приоритет запроса.

IOSB - Двухсловный буфер, куда помещается код завершения запроса.

AST - Адрес подпрограммы асинхронного подпроцесса.

ADR - Адрес блока позиционно-независимых кодов в программе.

SIZE - Размер блока кодов не должен превышать 7168 байт.

TIM - В младшем байте содержится таймаут запроса. (Поможет избежать зависания запроса 'навечно'.)

Требования к написанию блока кодов:

- 1). Блок позиционно-независимых кодов не должен загружать процессор на время более 500 мкс.
- 2). Размер блока кодов не должен превышать 3584 слов.
- 3). Запрещается использовать регистры R6 и R7.
- 4). В начале блока кодов нужно спасти регистры, т.к.:
 - R0 - Содержит адрес возврата в драйвер.
 - R4 - Содержит адрес SCB.
 - R5 - Содержит адрес UCB.
- 5). В конце блока кодов необходимо восстановить регистры: R0, R4, R5.
- 6). В регистр R1 помещается код завершения запроса.
- 7). Последней командой блока кодов является возврат в драйвер: JMP (R0).
- 8). За командой возврата может находиться буфер данных.

Пример оформления блока кодов:

```
MOV R0,XPC; save address of return to driver.
```

```
MOV R4,SAVE4; save R4 (SCB).
```

```
MOV R5,SAVE5; save R5 (UCB).
```

```
... ..
```

```
MOV #IS.SUC,R1; successfull finish.
```

```

MOV SAVE4,R4; restore R4.
MOV SAVE5,R5; restore R5.
JMP (R0): return to driver.
XPC:. WORD 0; BUFFER for data.
SAVE4:.WORD 0;
SAVE5:.WORD 0;

```

1.2.4. Алгоритм работы драйвера

Драйвер CCDRV работает с одним DCB, на каждый крейт создается по одному SCB, UCS. Каждому крейту соответствует логическое имя:

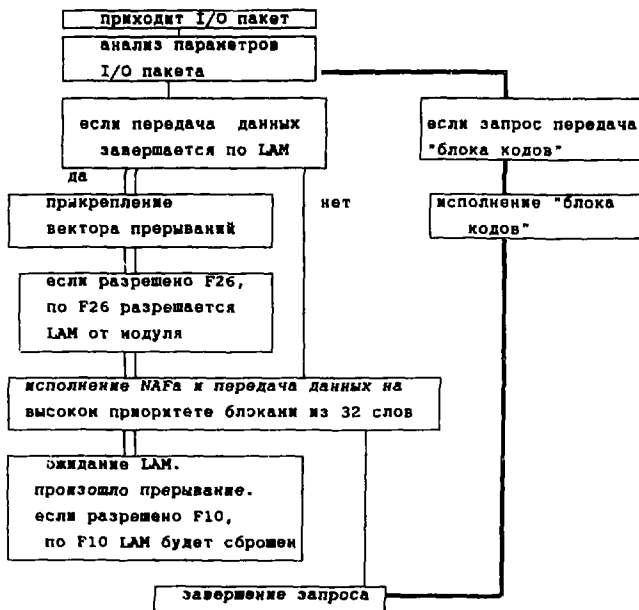


Рис. 4. Блок-схема KANAK драйвера CCDRV: первому - CC0;, второму - CC1, и т.д. [4].

Классифицируются три типа запросов к драйверу:

- 1). Запрос на передачу данных, завершающийся проверкой 'X', 'Q'.
- 2). Запрос на передачу данных, завершающийся обработкой прерывания, при выставлении модулей "LAMA".
- 3). Запрос на передачу блока кодов (функция IO.LOD).

На Рис. 4 указаны три возможных метода обслуживания запроса.

1.3. САМАС-АСР

RSX-11M - это многозадачная система, в которой с периферийной аппаратурой могут работать одновременно несколько пользователей. Значит возможна ситуация, когда в одном крейте стоят два модуля, с каждым из них работают две разные задачи. Требуется иметь возможность выполнить из задачи некую процедуру, которая бы позволила защитить модуль от "чужой задачи". Ancillary Control Processor САМАС-АСР выполняет функции контроля и защиты модулей крейта (аналогично файловому процессору FILES11-АСР). Задача захватывает модуль и устанавливает код доступа. Если установлена защита на запись и чтение, то из других задач нельзя ни прочитать, ни записать данные в модуль. С помощью задачи ССИ монтируется крейт, после чего все запросы к этому крейту контролирует САМАС-АСР. ССИ выполняет операции монтирования и демонтажирования крейтов, посылает Z в крейтконтроллер, переводит крейт в состояние "online" или "offline".

1.3.1. Запрос на присвоение LUN модулю и установка защиты

Эта процедура присваивает LUN модулю и устанавливает код защиты модуля от доступа из "чужих задач".

QIOW\$ IO.ACE.LUN,EVENT,PRI,IOSB,AST,<,,MODUL,WORD,,>

LUN - Логический номер, назначаемый модулю.

EVENT - Флаг завершения процесса.

PRI - Приоритет запроса.

IOSB - Адрес двухсловного блока, куда помещается код завершения запроса.

AST - Адрес подпрограммы асинхронного подпроцесса.

MODUL - Номер позиции модуля в крейте задан в поле с девятого по пятый бит [N16 - N1].

15		9	8	7	6	5	4	0
		N16	N8	N4	N2	N1		

WORD - Содержит код защиты:

WI.LCK=004000 Модуль доступен для записи и чтения только из задачи, прикрепившей его.

WI.RDV=001000 Модуль доступен для чтения только из задачи, закрепившей его. Другая задача может прикрепить его без защиты и писать данные в него.

WI.WDV=000400 Модуль доступен для записи только из задачи, закрепившей его. Другая задача может прикрепить его без защиты и читать данные из его.

Если **WORD** содержит ноль, то модуль будет прикреплен без защиты, другие задачи могут обмениваться данными с ним в обоих направлениях.

1.3.2. Запрос на снятие защиты с модуля

Ниже описанная процедура позволяет снимать защиту с модуля на чтение или запись данных из "чужих задач", и откреплять модуль.

`QIOW$ IO.DAC,LUN,EVENT,PRI,IOSB,AST,<,, ,WORD,,>`

LUN - Логический номер, назначенный модулю.

EVENT - Флаг завершения запроса.

PRI - Приоритет запроса.

IOSB - Адрес двухсловного буфера, куда помещается код завершения запроса.

WORD - Код снятия защиты:

WI.LCK=004000 Защита на чтение и запись снимается, и модуль открепляется от задачи.

WI.RDV=001000 Защита на чтение снимается.

WI.WDV=000400 Защита на запись снимается.

1.3.3. Запрос на чтение или запись виртуального блока

При обмене данными с модулем, находящемся в монтированном крейтке используется термин - виртуальный блок, в запросе посылается в крейт **NAF**, и происходит обмен данными.

Запрос на чтение данных:

```
QIOW$ IO.RVB,LUN,EVENT,PRI,AST,<ADR,SIZE,TIM,WORD,,>
```

Запрос на запись данных:

```
QIOW$ IO.WVB,LUN,EVENT,PRI,AST,<ADR,SIZE,TIM,WORD,,>
```

В 1.2.1 и 1.2.2 описаны функциональные значения параметров запросов. В параметре WORD значенке поля с пятого по девятый бит игнорируется.

1.4. ОПИСАНИЕ БИБЛИОТЕКИ DAS

1.4.1. Подпрограммы для работы с КАМАК

Библиотека подпрограмм позволяет пользователю на FORTRANе писать программы для работы с КАМАК аппаратурой, выполняющие управление и контроль, и сбор данных. Задача пользователя может работать с немониторинговым крейтом или с мониторинговым (1.3). Утилита CCI выполняет мониторинг крейта, после этого САМАС-АСР контролирует все запросы, приходящие к модулям этого крейта. В начале описание подпрограмм позволяющих работать с немониторинговым крейтом.

1). Нужно прикрепить крейт к задаче:

```
CALL ATTACH (NCRATE, LUN, IOSB)
```

NCRATE - Номер крейта.

LUN - Логический номер, присвоенный крейту.

IOSB - Код завершения операции.

2). Следует декларировать номер позиции модуля в крейте. Каждый раз, когда начинаете работать с каким-либо модулем. Сначала декларируйте позицию, а затем производите обмен данными между модулем и задачей.

```
CALL LAM (MODUL)
```

MODUL - Номер позиции модуля в крейте.

3). Подпрограмма для чтения данных из КАМАКА:

```
CALL IORVB (LUN, NBUFFER, NSIZE, NFUN, NSUBADR, I24, NQ, IOSB)
```

Подпрограмма для записи данных в КАМАК:

```
CALL IOWVB (LUN, NBUFFER, NSIZE, NFUN, NSUBADR, I24, NQ, IOSB)
```

LUN - Логический номер, назначенный крейту.
NBUFFER - Адрес буфера данных.
NSIZE - Размер передаваемых данных в байтах.
NFUN - Код функции [F32 - F0].
NSUBADR - Субадрес [A16 - A0].
I24 - Если I24 = 1, тогда передача идет двадцатичетырех-битовыми словами. Если I24=0, то передача идет шестнадцатичетырехбитовыми словами.
NQ - Служебный параметр (NQ=1).
IOSB - Код завершения процедуры.

4). После завершения работы с модулями крейта, нужно открепить крейт:

CALL DETACH (LUN, IOSB)

LUN - Логический номер, присвоенный крейту.
IOSB - Код завершения процедуры.

Для работы с монтированным крейтом необходимо выполнять процедуры в следующей последовательности:

1). Нужно назначить LUN модулю крейта и установить защиту:

CALL LUNCC (NCRATE, MODUL, LUN, IPROTECT, IOSB)

NCRATE - Номер крейта.
MODUL - Номер позиции модуля в крейте.
LUN - Логический номер, назначенный модулю.
IPROTECT - Код защиты модуля (в 1.3.1 подробно описана защита):

'RO' - защита на запись
 'WO' - защита на чтение
 'RW' - защита на чтение и запись
 'OO' - нет защиты ни на чтение ни на запись.
IOSB - Код завершения процедуры.

2). Обмен данными между задачей и модулем.

Чтение данных из модуля:

CALL IORVB(LUN, NBUFFER, NSIZE, NFUN, NSUBADR, I24, NQ, IOSB)

Запись данных в модуль:

CALL IOWVB(LUN, NBUFFER, NSIZE, NFUN, NSUBADR, I24, NQ, IOSB)

3). После завершения работы с модулем нужно открепить модуль или снять защиту:

CALL UNLOCK (LUN, IPROTECT, IOSB)

LUN - Логический номер, присвоенный модулю.
IPROTECT - Код защиты модуля:
 'RO' - Снять защиту на чтение.
 'WO' - Снять защиту на запись.
 'RW' - Снять защиту на чтение и запись.
 'EX' - Открепить назначенный LUN, т.е. закрыть логический канал, связывающий задачу и модуль.
IOSB - Код завершения процедуры.

1.4.2. Подпрограммы для работы с буферным банком и архивом

Файлы DAS: буферный банк и архив имеют структуру, недоступную для обычных FORTRANовских процедур работы с файлами. Для работы с файлами написаны подпрограммы, посредством которых можно открыть файл, прочитать или записать в него данные. DASовский механизм работы с файлами сокращает время доступа к данным, и не подбивает к задаче много дополнительной памяти, как это делают FCS или RMS.

Открыть файл, доступный для чтения и записи:

CALL OPENFL (LUN, 'DDNN: [NNN, LLL] NAME. TYP; VER' ,ICODE, IOSB)

LUN - Логический номер, назначенный файлу.
'DDNN: [NNN, LLL] NAME. TYP; VER' - В апострофах заключен текст, где указано полное имя файла, это может быть и текстовая переменная:

DDNN: - Логическое имя диска.
[NNN,LLL] - UIC пользователя.
NAME - Имя файла.
TYP - Тип файла.
VER - Версия файла.
ICODE - Текстовая переменная или непосредственно текст, определяющий способ открытия файла:
'NEW' - Создание нового файла.
'OLD' - Открытие уже существующего файла.

'UNKNOWN' - Открытие уже существующего файла, если такого нет, то создание нового.

IOSB - Код завершения процедуры.

Открыть файл, доступный только для чтения:

CALL OPENRO (LUN, 'DDNN: [NNN, LLL] NAME. TYP;VER', ICODE, IO SB)

Чтение массива данных из буферного банка происходит по двум ключевым словам: UIC пользователя и идентификатор данных. Каждому массиву данных соответствует идентификатор - имя из шести букв, которое является опознавательным словом.

Чтение данных из буферного банка:

CALL READB (LUN, IUIC, IDENT, NSIZE, BUFFER, NCRATE, MODUL, IO SB, ITIME)

Входные параметры:

LUN - Логический номер, назначенный файлу.

IUIC - UIC пользователя, в старшем байте находится номер группы, а в младшем находится номер члена группы. Если UIC равен нулю, тогда не контролируется UIC данных. Если UIC пропущен, тогда контроль производится по текущему UIC.

IDENT - Идентификатор данных (до шести символов ASCII).

NSIZE - Размер читаемых данных в байтах. А выходное значение NSIZE равно размеру данных, находящихся в файле под этим идентификатором.

Выходные параметры:

BUFFER - Адрес буфера читаемых данных.

NCRATE - Номер крэйта.

MODUL - Номер позиции модуля (для raw-data).

Если данные занесены в буферный банк пользователем, то на выходе NCRATE=-1, MODUL=-1 (для user-data).

IO SB - Код завершения процедуры.

ITIME - Адрес буфера, куда помещаются 18 символов - время и дата выстрела (необязательный параметр).

Чтение данных из архива происходит по шести ключевым словам: имя компьютера (это имя узла DECnet, если DECnet не загружен, тогда имя компьютера - RSX11M), дата и время эксперимента, серия экспериментов, номер эксперимента, UIC пользователя, идентификатор данных.

Чтение данных из архивного файла:

CALL READA (LUN, COMP, DATE, BACKUP, NUMBER, IUIC, IDENT, NSIZE,
BUFFER, NCRATE, MODUL, IOSB, ITIME)

Процедура чтения из архива отличается от процедуры чтения из буферного банка только дополнительными четырьмя параметрами:

COMP - Имя компьютера, где были собраны данные, 32 символа.

DATE - Дата и время сбора данных, 18 символов.

BACKUP - Имя серии экспериментов, 20 символов.

NUMBER - Номер эксперимента.

Пользователь может внести в буферный банк обработанные данные (user-data) под неким идентификатором.

CALL WRITEB (LUN, IUIC, IDENT, NSIZE, BUFFER, IOSB) LUN - Логический номер, назначенный файлу.

IUIC - UIC пользователя.

IDENT - Идентификатор данных, вносимых в буферный банк. (6 символов ASCII).

NSIZE - Размер данных в байтах.

BUFFER - Адрес массива данных.

IOSB - Код завершения процедуры.

2. РЕДАКТОР БУФЕРНОГО БАНКА

Редактор буферного банка EDITAC позволяет создавать и редактировать описатели КАМАК модулей и алгоритмы их работы. Для поддержки каждого КАМАК модуля создается подпрограмма, ведущая диалог, в ходе которого задается режим работы модуля и описание алгоритма его работы. В меню редактора указан список КАМАК модулей, для которых уже имеется подпрограммная поддержка. Структура буферного банка такова: в первых блоках файла находятся описатели модулей, далее следует область описания алгоритмов работы КАМАК аппаратуры, далее идут описатели данных, прочитанных из КАМАК модулей (raw data), потом следует информация, собранная входе эксперимента. В первом блоке буферного банка содержится описание структуры файла (адрес поля и его размер указаны в байтах):

адрес поля	размер	значение поля
B\$MCRT =0	2	максимальный номер крейта.
B\$LAST =2	2	номер последнего занятого блока в файле
B\$CSHT =4	2	номер блока, где находится комментарий к выстрелу.
B\$DSCI =6	2	номер блока, где находится область описателей собранных данных (raw-data).
B\$NUM1 =10	2	число описателей raw-data.
B\$USE1 =12	2	номер блока, где находится область описателей данных (user-data), внесенных в банк пользовательскими задачами.
B\$USNT =14	2	число описателей user-data.
B\$CC00 =16	2	в этом слове отображается логическое состояние крейтов. Номер бита соответствует номеру крейта. Если бит установлен, то крейт в состоянии 'online', если бит погашен - крейт в состоянии 'offline'.
B\$COMP =340	32.	32 символа ASCII - имя компьютера (узла)
B\$SERS =400	20.	20 символов ASCII - имя серии выстрелов.
B\$DATA =424	18.	18 символов - дата и время выстрела.
B\$SHOT =446	2	номер выстрела.
B\$FNH =450	2	номер последнего блока области, где описаны алгоритмы работы КАМАК модулей.
B\$ARH =776	2	режим архивирования банков для EXCGDL -1 - EXCGDL автоматически сбрасывает банки в архив 0 - указания на архивирование принимаются с терминала 1 - архивирование не производится

Начиная со второго блока, расположена область описателей модулей КАМАК. Размер этой области определяется максимально возможным числом крейтов, которое задается при создании банка. В области описателей на каждый модуль заводится запись, местонахождение ее определяется номером описателя NDSC, где CRATE - номер крейта (от 0 до максимального), MODUL - номер позиции модуля в крейте (от 1 до 23):

$$NDSC = CRATE*23 + MODUL .$$

Если в поле номера описателя находится 0 это означает, что либо нет описания алгоритма работы модуля, либо модуль в состоянии "offline". В записи (описателе модуля) имеются следующие поля:

адрес поля (номер байта)	размер поля (в байтах)	значение поля
L\$DSC =0	2	номер описателя.
L\$LDLH =2	2	резерв.
L\$CRCT =4	1	номер крейта.
L\$MOD =3	1	номер модуля.
L\$UIC =6	2	код UICa.
L\$PRI =10	1	резерв.
L\$FLAG=11	1	резерв.
L\$APRE=12	2	адрес области описания алгоритма подготовки к выстрелу.
L\$APOS=14	2	адрес области описания алгоритма сбора информации.
L\$COMM=16	2	адрес комментария к модулю.
L\$NAME=20	10.	имя модуля в кодах ASCII.

Имеется две области описания алгоритмов. Разделение их на алгоритмы подготовки к выстрелу и алгоритмы сбора данных условно, обе области имеют одинаковую структуру, только исполняемая задача EXCGDL различает их. Адресом алгоритма является номер блока. Если в поле адреса содержится 0, это означает, что алгоритм отсутствует. Комментарий к модулю содержит некий текст, поясняющий функции данного модуля. Область описания алгоритма представляет собой список описаний запросов к крейту:

адрес поля	размер	значение поля
P\$NLUN=0	2	номер описателя модуля.
P\$LUN =2	1	резерв.
P\$PRG =3	1	1 - если код запрос 'IO.LOD' 0 - если коды запросов: IO.RVB или IO.WVB.
P\$MOD =4	1	номер позиции модуля в крейте.
P\$NAP =5	1	число NAFов, равное числу описателей запросов к данному модулю в крейте.
описание запроса:		
P\$ANEX=6	2	размер в байтах области запроса.
P\$ONEX=0		(смещение относительно области запроса).
P\$F =10	1	код КАМАК функции.
P\$2F =2		(относительное смещение)
P\$A =11	1	код субадреса.
P\$3A =3		(относительное смещение)
P\$RW =12	2	код запроса IO.RVB (чтение данных из модуля) или IO.WVB (запись данных в модуль).
P\$4RW =4		(относительное смещение)
P\$BITS=14	2	слово содержит параметры, дополняющие функцию запроса (в 1.2.1 они описаны).
- -		(относительное смещение)
P\$6BIT=6	2	
P\$SIZE=16	2	размер массива передаваемых данных в байтах
P\$10S2=10		(относительное смещение)
P\$DATA=20	6	если запрос на чтение данных, это поле содержит идентификатор массива прочитанных данных. Если запрос на запись данных, то в этом поле находится массив передаваемых данных.
P\$12DT=12		(относительное смещение)

EDITAC - редактор буферного банка предоставляет в меню следующие возможные операции:

Выход из EDITAC	=> E
Редактирование алгоритма работы модуля	=> 1
Перевести модуль в состояние 'online/offline'	=> 2
Каталог описанных модулей	=> 3
Перепись вашего банка в общий буферный банк	=> 4
=>	

EDITAC переводит модули и крейты в состояния "online" или "offline". Если модуль или крейт в состоянии "offline", EKSGDL алгоритмы для этой аппаратуры пропускает.

Операции редактирования алгоритма работы модуля позволяет, либо создать алгоритм, либо изменять параметры режима работы. Вы задаете номер крейта и номер позиции модуля, и выходите на меню, где указаны имена КАМАК модулей, включенных в EDITAC. Выбираете модуль и задаете параметры режима работы, а соответствующая подпрограмма преобразует их в последовательность NAFов и создает в буферном банке алгоритмы работы модуля.

Например, для АЦП-101SK [6] диалог такой:

```

      Режим работы:
Внешний единичный запуск - 0
Внешний многократный запуск - 1
--> 0
Временной квант' АЦП 1 мксек [Yes/Next/Back] ----> Y
Число сканируемых каналов [1; 2; 4] ----> 1
Номер канала [0; 1; 2; 3] ----> 0
Диапазон измерения напряжений 1.28 в [Yes/Next/Back] --> Y
Идентификатор данных [до 8 символов] ----> CHANAL
Размер данных [до 3966 слов] ----> 2048
  
```

Выбрали "внешний единичный запуск", временной квант измерения установили 1 мксек, сканирование одного канала. Установили для нулевого канала диапазон напряжений от - 1.28 в до + 1.28 в, и читаем 2048 слов, которые в буферном банке образуют массив под именем CHANAL.

EDITAC написан на FORTRANе, что позволяет легко модифицировать редактор буферного банка, и вставлять подпрограммы для поддержки работы новых КАМАК модулей. Ниже описан механизм включения в EDITAC нового модуля.

- 1). Создаете подпрограмму, которая ведет диалог. Она принимает номер крейта и позицию модуля в крейте, а также определяет имя модуля, затем принимает параметры режима работы.
- 2). Далее создаете описание алгоритма работы модуля на основе принятой информации с помощью следующих подпрограмм:

I. Открываете буферный банк, используя LUN равный

1. CALL OPENFL(LUN, "SY:CAMAC. DCS", "OLD", IOSB)

II. Параметры модуля задаете в коммон-блоке "PROGRAM".

COMMON [PROGRAM] IUIC, LUN, ICRATE, MODUL, NAME(5)

IUIC - UIC пользователя.

LUN - LUN равен 1.

ICRATE - Номер крейта.

MODUL - Номер позиции модуля в крейте.

NAME(5) - Имя модуля (до 10 символов ASCII).

III. Создаете описатель модуля.

CALL DSCRPT(IOSB)

Если IOSB равен 1, это значит - создан новый описатель модуля. Если IOSB равен 2 - открыт существующий описатель модуля. Если IOSB меньше или равен - 1, произошла ошибка при выполнении операции.

IV. Далее некую область в банке резервируете под описание алгоритма.

CALL ALGRTR(MODA, NUMNAF, IOSB)

MODA - Если создаете алгоритм подготовки к выстрелу, тогда MODA=0. Если алгоритм сбора данных - MODA=-1.

NUMNAF - Число NAFов в алгоритме.

IOSB - Код завершения операции.

V. Создаете логическую цепочку NAFов. Если NAF выполняется запись данных в КАМАК модуль, то используете подпрограмму:

CALL WNAF(BUFFER, ISIZE, IFUN, ISADR, I24, IWAIT, IOSB)

BUFFER - Массив данных, которые передаются в модуль.

ISIZE - Размер данных в байтах.

IFUN - Код функции (от F0 до F12).

ISADR - Субадрес (от A0 до A24).

I24 - Если I24 = 0-передача идет шестнадцатититовыми словами.

Если I24 = 1 - двадцатититовыми словами.

IWAIT - Служебный параметр, равный 1.

IOSB - Код завершения операции.

Если NAF выполняет операцию чтения данных из модуля в банк, тогда используете подпрограмму:

CALL RNAF (NAME, ISIZE, IFUN, ISADR, I24, IWAIT, IOSB).

NAME - Идентификатор данных, прочитанных из КАМАК (из шести символов). Это имя будет использовано при чтении данных из буферного банка.

Назначение остальных параметров совпадает со списком параметров **WNAF**.

VI. Далее используете операцию закрытия описателя алгоритма:
CALL CLSALG

VII. Если нужен комментарий, то используйте подпрограмму: **CALL COMMEN**

VIII. Самой последней следует процедура закрытия описателя модуля:

CALL CLSDSC

IX. Если читаете какой-нибудь **NAF** в описании алгоритма работы модуля, то вначале определите координаты банка в файле. (Процедура чтения **NAF**а может быть выполнена как для буферного банка, так и для архивного файла.) Если чтение происходит из архивного файла, то вначале выполните процедуру позиционирования:

CALL HOLD(LUN, NODE, IDATA, IBACKUP, ISHOT, IOSB)

LUN - LUN, назначенный архивному файлу.

NODE - Имя компьютера (32 символа ASCII).

IDATA - Дата и время выстрела (18 символов).

IBACKUP - Имя серии выстрелов (20 символов).

ISHOT - Номер выстрела.

IOSB - Код завершения процедуры.

Если читаете из буферного банка, то координаты банка укажите в коммон-блоке "**HOLDBN**":

COMMON [HOLDBN] NBANK, NSIZE, IFLAG

NBANK - Номер блока, с которого начинается банк.

NSIZE - Размер банка в блоках.

IFLAG - флаг.

(Всем этим параметрам присвойте значение 1.)

X. Если вы измените какой-то **NAF** в существующем алгоритме работы **КАМАК** модуля, вначале прочитайте его:

CALL RNAFR (NCR, MOD, NAME, MODA, NNAF, IFUN, ISADR, IO, I24, IWAIT, IF26, IF10, INTRT, ISIZE, ISTORE, IOSB)

Входные параметры:

NCR - Номер кредита.

MOD - Номер позиция модуля в крейте.
NAME - Имя модуля (до 10 символов).
MODA - Если MODA = 0, алгоритм подготовки к выстрелу. Если
MODA = - 1, алгоритм сбора данных.
NNAF - Номер читаемого NAF.

Выходные параметры:

IFUN - Код функции.
ISADR - Субадрес.
IO - Код операции. Массив из четырех байт. "Read" - чтение,
"writ" - запись.
I24 - Размер КАМАК-слова. Если I24 = 0 - обмен идет шест-
надцатибитовыми словами; если I24 = 1 - передача идет
двадцатибитовыми словами. IWAIT - Служебный параметр
равен 1.
IF26 - Если IF26 = 1 - перед исполнением NAF выполняется
F26.IF10 - Если IF10 = 1 - перед завершением запроса по F10
сбрасывается LAM.
INTRT - Если INTRT =1, организуется процесс обработки прерывания.
ISIZE - Размер передаваемых данных в байтах.
ISTORE - Буфер. Если операция чтения, тогда буфер содержит
идентификатор данных. Если операция записи, тогда в
буфере содержатся данные, занесенные в КАМАК.
IOSB - Код завершения операции.

XI. С помощью подпрограммы WNAFW запишите новый NAF:

CALL WNAFW(NCR, MOD, NAME, MODA, NNAF, IFUN, ISADR, IO, IWAIT,
IF26, IF10, INTRT, ISIZE, ISTORE, IOSB)

Если вы создаете программную поддержку для КАМАК модуля,
который помещает собранные данные в буферный банк, напишите под-
программу для чтения параметров режима его работы, чтобы можно
было перевести прочитанные данные в обычные физические величин-
ны. Например: для КАМАК модуля АЦП-101СК [6] уже имеется под-
программа:

CALL RACPF(LUN, NCRATE, MODUL, KANAL, ISTATS, COMMUN, TIMEQ,
MEMORY, NAME, IOSB)

Входные параметры:

LUN - LUN буферного банка.
NCRATE - Номер крейта.

MODUL - Позиция модуля в крейте.
Выходные параметры:
KANAL - Число сканируемых каналов.
ISTATS - Код статуса.
COMMUN - Массив из четырех реальных чисел, соответствующих амплитудным квантам каналов.
TIMEQ - Временной квант измерения.
MEMORY - Размер памяти: для АЦП-101SK - 4096 слов, для АЦП-850SK [6] - 1024 слов.
NAME - Массив из 12 слов, предназначен для четырех имен каналов.
IOSB - Код завершения процедуры.

С помощью RACPF определяете параметры режима работы, из банка или архива читаете массив данных в кодах АЦП-101SK, и далее можно производить отображение информации в виде осциллограммы.

3. ИСПОЛНЯЮЩАЯ ЗАДАЧА

Работу с КАНАК аппаратурой ведет утилита EXCGDL. Основные ее функции: из буферного банка считать алгоритмы работы с КАНАКом, преобразовать их в последовательность запросов к КАНАК модулям. Если есть прочитанные данные (raw data), то занести их в буферный банк, зафиксировать время, дату, серию и номер эксперимента. Блок-схема на Рис. 5 дает представление о структуре EXCGDL. При работе с КАНАК аппаратурой EXCGDL контролирует состояние крейтов и модулей. Если крейт или модуль логически или физически отключен, тогда никакие операции с ним EXCGDL не выполняет. Если при выполнении запроса происходит сбой, то на терминал выдается сообщение об ошибке: номер крейта, номер позиции модуля, код функции, код ошибки. Утилита дополнительно выполняет операции: занесения буферного банка в архив; запуск задач, производящих анализ и контроль работы диагностических систем. (В REQUEST.JOBс помощью редактора EDITAC помечено, какие задачи запускать входе текущего эксперимента).

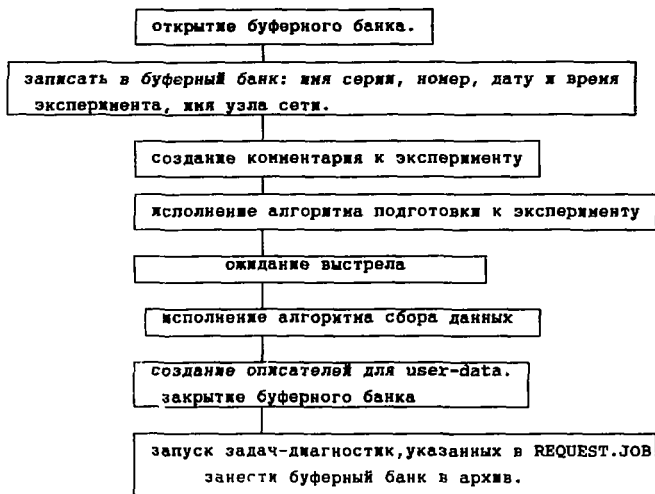


Рис. 5. Блок-схема управляющей утилиты EXCGDL.

4. АРХИВ

Утилита ARCHIV выполняет все операции архивирования Рис. 6. Обычно утилита сбрасывает буферные банки в один файл – архив, ведёт бортовой журнал, где фиксирует имя серии, номер эксперимента, дату, время и полное имя архивного файла (т.к. предусмотрена возможность иметь сразу несколько архивных файлов). ARCHIV извлекает любой буферный банк из архива; выдаёт каталог архивного файла; распечатывает протокол архивирования, содержащийся в журнальном файле. Из пользовательских задач можно простым способом прочитать любой массив данных: вы открываете архивный файл к в списке параметров подпрограммы чтения данных указываете серию, номер выстрела и идентификатор данных. Нет ограничений на объём и тип данных, находящихся в буферном банке или архиве.

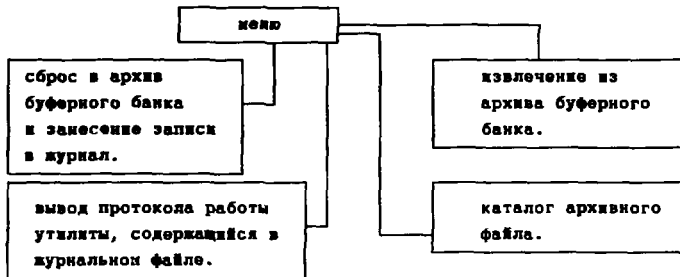


Рис.6 Блок-схема утилиты ARCHIV.

описание структуры файла
каталог банков
банк 1
...
банк N
продолжение каталога банков
банк N+1
...

Рис. 7. Структура архивного файла.

В первом блоке архива содержится описание структуры файла:

Адрес поля (номер байта)	размер поля (в байтах)	значение поля
A\$NUMB = 0	2	число буферных банков в архиве.
A\$LAST = 2	2	номер последнего занятого блока в файле.
A\$KATL = 4	2	число блоков, занятых под каталог банков.
A\$LSTK = 6	2	номер последнего блока каталога.
A\$SQC =20	2	начиная с этого поля следуют номера блоков каталога.

Каталог состоит из блоков, в каждом блоке содержится 12 записей. В 478-й байте блока находится номер следующего блока

адрес поля (номер байта)	размер поля (в байтах)	значение поля
K\$NUM =0	2	номер записи.
K\$SER =2	20.	20. символов ASCII - имя серии выстрелов.
K\$SHOT =26	2	номер выстрела.
K\$DATA =30	18.	18 символов ASCII - дата и время выстрела.
K\$BLCK =52	2	номер блока, где находится начало банка.
K\$SIZE =54	2	размер банка в блоках.
K\$COMM =56	2	номер блока, где находится комментарий к выстрелу в данном банке.

каталога. Если это последний блок каталога, то 478-й байт содержит ноль. Каталог состоит из записей следующего формата:

Возникает вопрос - "зачем нужно было создавать свою базу данных? "Доступные базы данных (дистрибутивы которых мы имели): ORACLE, DATATRIEVE и ADABAS неприемлемы. Т.к. время доступа к данным порядка от 2 до 8 минут, объемы передаваемых данных ограничены, и очень сложный механизм доступа к данным. Избранный метод архивирования позволяет: во-первых, очень простым способом получить к данным. Во-вторых, нет ограничений на объемы передаваемых данных и их структуру. В-третьих, время доступа 2 Кслов/сек для данных, хранящихся на диске винчестер фирмы "ST-225 SeaGate". Для файлов, находящихся на электронном диске DE: время доступа равно 11 Кслов/сек. В-четвертых, архивный файл, как из кубиков, создается из буферных банков. Возможен и обратный процесс: из архива извлекается буферный банк, содержащий полную информацию о ходе эксперимента. Банки хранятся в архиве без изменения их структуры.

5. СРАВНИТЕЛЬНЫЙ ОБЗОР СИСТЕМ КОНТРОЛЯ И СБОРА ДАННЫХ

На основе DAS построена система контроля, управления и сбора данных Рис. 8 для установки ГДЛ. С помощью диалогового редактора MNGGDL в буферном банке WORK.DCS создается описание режима работы аппаратуры задаются параметры: а) плазменного генератора; б) системы зарядки конденсаторных батарей; в) выдача

стартовых импульсов для инжекторов, диагностической аппаратуры и т.д. Задача GDLMNG исполняет алгоритмы, заданные в WORK.DSC, и производит контроль и управление работой установки; если нужно, запускает задачу EXCGDL, выполняющую сбор данных. Из EXCGDL происходит инициация задач, которые выполняют обработку собранной информации и выводят на мониторы ЦДР-2 [7] характеристики плазмы. Например: зависимость концентрации и температуры плазмы от времени и т.д. EXCGDL помещает банк CAMAC.DSC с собранной информацией в архив ARCHIV.DSC. В интервале между выстрелами экспериментаторы имеют возможность произвести анализ и контроль работы диагностических систем. Это позволяет контролировать и менять параметры оборудования входе эксперимента.

Исторически в ИЯФ сложилась двухуровневая распределенная система автоматизации на плазменных установках [8], [9]. Когда на верхнем уровне находился host компьютер с ОС RSX-11M, а на нижнем - target компьютер с ОС RSX-11S, связанные сетью АЛИСА. Либо в target компьютере работала без ОС задача (FORTRAN stand-alone). В target машине происходила работа с КАМАКом, а host компьютер передавались собранные данные, где происходила их обработка и архивирование [10], [11]. Ранее на ГДЛ использовалась двухуровневая система, которая требовала сложной аппаратуры

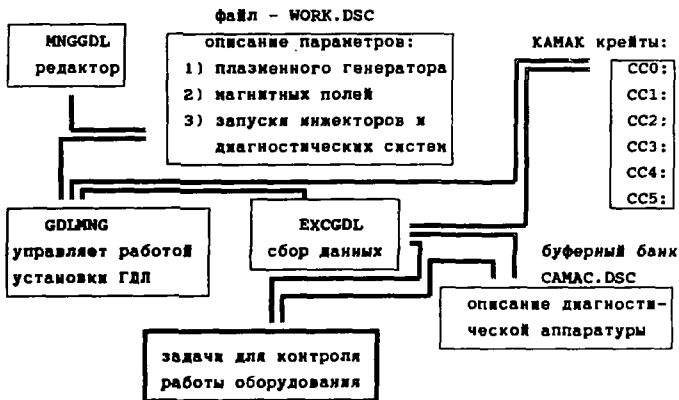


Рис.8. Схема управления и сбора данных установки ГДЛ.

и программной поддержки, что снижало время реакции системы и ее надежность. Низкая производительность системы в интервале между выстрелами позволяла экспериментаторам только прочитать в своих target машинах данные из КАМАКа и поместить в архив. А после завершения серии экспериментов, производился анализ собранных данных, что не позволяло оперативно корректировать работу оборудования. При создании проекта DAS-ГДЛ были просмотрены публикации на тему "Системы сбора данных", более привлекательной по своим возможностям является MDS-MIT [12], которая используется на плазменных установках в лабораториях: Plasma Fusion Center, Massachusetts Institute of Technology [13]; Plasma Physics Laboratory, Princeton University [14]. MDS работает на компьютерах VAX/VMS. DAS-ГДЛ работает на "Электрониках MC1212". Производительность компьютеров VAX и более совершенная аппаратура, используемая системой MDS, делает ее комфортабельной для экспериментатора; но DAS позволяет решать те же задачи и на более доступной отечественной аппаратуре. DAS можно использовать для ведения экспериментов не только в области физики плазмы, но и для автоматизации научных работ или технологических процессов на оборудовании, управляемом КАМАКом.

Проект DAS был разработан для установки ГДЛ совместно с П.А. Багрянским и В.В. Клесовым. Запуск и отладка DAS происходили при участии П.А. Багрянского и А.Н. Карпушова. Я очень признательна им за участие в этой работе!

Я благодарна А.Н. Кирпотину за конструктивные замечания, сделанные в ходе прочтения текста статьи!

ЛИТЕРАТУРА

- 1 В.И. Давыденко, А.А. Иванов, Ю.Л. Козьминых и др. "Экспериментальная модель газодинамической ловушки." Препринт ИЯФ СО АН 866-104, Новосибирск 1986.
- 2 "Overview of DECnet." Документация DEC.
- 3 "RSX-11M/M-PLUS I/O Driver Reference Manual." Документация DEC.
- 4 "Writing to I/O driver." Документация DEC.
- 5 В.И. Нифонтов, А.Д. Орешков, А.Н. Путьмаков и др. "Контроллер и драйвер для организации связи в последовательном входе между ЭВМ "Электроника-60" и крейтами КАМАК." Препринт ИЯФ СО АН 882-90. Новосибирск 1982.

- 6 М. Вайраков, В.Р. Козак, В.И. Нифонтов и др. "Цифровые осциллографические модули в стандарте КАМАК." В сборнике трудов "2-го Всесоюзного симпозиума по модульным информационно вычислительным системам." Москва, 1980.
- 7 Э.А. Кулер, В.И. Нифонтов, Г.С. Паскунов, В.В. Репков. "Цветной графический дисплей." В книге "Автоматизация научных исследований на основе применения ЭВМ." Тезисы докладов 1-й Всесоюзной конференции. Новосибирск, 1979.
- 8 А.В. Проскурин. "Многомашинный комплекс для управления крупными физическими установками на базе ЭВМ серии "Электроника". 2-й Всесоюзный семинар по автоматизации научных исследований в ядерной физике и смежных областях. Новосибирск, 1982.
- 9 Н.В. Бейлин, О.В. Вьюжин, А.Д. Клименко и др. "Многомашинный комплекс автоматизации физического эксперимента." 2-й Всесоюзный семинар по автоматизации научных исследований в ядерной физике и смежных областях. Новосибирск, 1982.
- 10 В.В. Коняхов, Ю.А. Цибулко. "Пакет прикладных подпрограмм для автоматизации крупных физических установок на базе ЭВМ серии "Электроника". В книге "Обработка физической информации". Тезисы докладов 3-го Всесоюзного семинара по обработке физической информации. Цахкадзор. 9-12 октября 1984.
- 11 С.Г. Воронцов, В.В. Коняхов, К.И. Неклер и др. "Автоматизация и регистрации и обработки информации на установке "У-1" - генераторе мощного релятивистского электронного пучка для нагрева плазмы." В книге "Обработка физической информации" Тезисы докладов 3-го Всесоюзного семинара по обработке физической информации. Цахкадзор. 9 - 12 октября 1984.
- 12 T. W. Fredian, J. A. Stillerman. "MDS/MIT High-Speed Data Acquisition and Analysis Software System." Review of Scientific Instruments. Vol.57, N8, part 2, p.1907, August 1986.
- 13 J.D. Sullivan, M.P.J. Gaudreau, B. Rameriz, et.al. "The Tara Control, Monitoring, Data Acquisition and Analysis System." Massachusetts Institute of Technology PFC Report CP-86-12, September, 1986.
- 14 J. Murphy, T. Gibney. "PLT Data and Analysis System." Review of Scientific Instruments. Vol.57, N8, part 2, August 1986.

Т.В. Саликова

Система сбора данных установки ГДЛ

ИЯФ 82-42

Ответственный за выпуск С.Г. Попов

Работа поступила 16 июня 1992 г.

Подписано в печать 16.06. 1992 г.

Формат бумаги 60х90 1/16. Объем 2,2 печ.л., 1,8 уч.-изд.л.

Тираж 200 экз. Бесплатно. Заказ N 42.

Обработано на IBM PC и отпечатано на
роталпринте ИЯФ им. Г.И. Будкера СО РАН,
Новосибирск, 630090, пр. академика Лаврентьева, 11.