

индекс 3624



ЕРЕВАНСКИЙ ФИЗИЧЕСКИЙ ИНСТИТУТ

ЕФИ-801(28)-85

ЦЕНТРАЛЬНЫЙ НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
ИНФОРМАЦИИ И ТЕХНИКО-ЭКОНОМИЧЕСКИХ ИССЛЕДОВАНИЙ
ПО АТОМНОЙ НАУКЕ И ТЕХНИКЕ

Г. Е. БАБАЯН, А. Ж. КЕТИКЯН, А. А. ШАГИНЯН

СПЕЦИАЛИЗИРОВАННЫЕ ПОДПРОГРАММЫ-ФУНКЦИИ ДЛЯ
ПРОГРАММИРОВАНИЯ В СТАНДАРТЕ КАМАК ПОД УПРАВЛЕНИЕМ
КОНТРОЛЛЕРА ТИПА КК

ЕРЕВАН-1985

В работе [1] были описаны специальные функции, введенные в язык БЭЙСИК для удобства программирования в стандарте КАМАК на ЭВМ типа "Электроника-100/16" ("Электроника-100/25", СМ-4, СМ-3, ...). Эти функции были реализованы для крейт-контроллера ККИ6, поставляемого вместе с ЭВМ типа "Электроника-100/16".

В данной работе описывается пакет специальных подпрограмм-функций, реализованных на языке Макроассемблер и предназначенных для использования при программировании на языках ФОРТРАН и БЭЙСИК в стандарте КАМАК с крейт-контроллерами КК (КК60, ККПД) [2]. Необходимость создания данного пакета, наряду с теми факторами, которые приведены в работе [1], обусловлена двумя основными принципиальными различиями в устройствах крейт-контроллеров КК и ККИ6. Эти различия заключаются в способе обращения к модулям КАМАК и структуре системы прерываний.

ККИ6 имеет 8 адресуемых регистров: 4 регистра адреса ветви и секции, командный регистр (для МАК - команд), статусный регистр (для определения состояния крейта и управления им), регистр старшего байта, регистр младшего байта.

Обращение к конкретным модулям КАМАК производится только через эти регистры, то есть косвенно.

КК имеет 3 внутренних адресуемых регистра: регистр управления и состояния (RUS), регистр маски и запросов (RMZ), регистр старшего байта.

Контроллер использует непосредственную адресацию модулей и регистров КАМАК с линией адреса общей шины (ОШ).

В ККИ6 есть возможность организации двух видов прерываний: по запросу и ошибке. Разрешение того или иного прерывания осуществляется при установлении соответствующего разряда статусного регистра.

КК имеет более гибкую организацию структуры прерываний. Можно организовать обработку до восьми различных типов прерываний от различных станций (включая прерывание по ошибке X). Вместе с тем имеется возможность "маскировки" прерываний: разрешить или нет прерывание по той или иной линии (запрос от одной станции коммутируется на одну из восьми линий RMZ, и к одной линии могут быть скоммутированы запросы от произвольного количества станций).

Все это приводит к несовместимости их программного обеспечения, а также более эффективному и гибкому использованию систем электронных модулей стандарта КАМАК под управлением мини-ЭВМ посредством контроллеров типа КК.

Реализованные специальные подпрограммы-функции просты в обращении и позволяют пользователям на языке высокого уровня писать программы управления, сбора и накопления информации для любой сложной системы, созданной на базе мини-ЭВМ типа "Электроника-100/16" и модулей электроники в стандарте КАМАК. Следует отметить, что некоторые из подпрограмм-функций выполняют действия, которые реализуются двумя-тремя командами на языке

Макроассемблер. Они оформлены в виде отдельных подпрограмм-функций для создания полного и гибкого набора подпрограмм-функций.

Специальные подпрограммы-функции для работы в стандарте КАМАК

Каждая из специальных подпрограмм-функций по своему семантическому смыслу одинакова как для использования языка ФОРТРАН, так и языка БЭЙСИК. Однако их оформление и способ обращения как к независимой части программы в этих языках различны. В языке БЭЙСИК все подпрограммы оформлены в виде одной программы с именем СМС, имеющей различные входы. Каждый вход соответствует реализации одной конкретной функции, порядковый номер которой указывается в первом параметре обращения к СМС. Подпрограммы вызываются оператором CALL. В языке ФОРТРАН эти функции оформлены либо в виде подпрограммы, либо в виде функции и соответственно их вызов осуществляется оператором CALL, либо использованием имени подпрограммы-функции.

Ниже приводится описание и способ обращения к специальным подпрограммам-функциям, которые по своему назначению можно разделить на 4 класса.

Управление состоянием в магистрали крейта КАМАК

I. CALL CKLZ (СМС(I))^{*} - устанавливает 9-й разряд RUS, что приводит к генерации цикла Z (запуск-установка всех регистров крейта в начальное состояние) в магистрали крейта.

^{*} В скобках указывается вид обращения к данной функции из программы, написанных на языке БЭЙСИК.

2. CALL SICLC (CMC(2)) устанавливает 8-й разряд RUS , что вызывает генерацию цикла C (общий сброс всех регистров крейта) в магистрали крейта.
3. CALL SICLI (CMC(3)) - устанавливает 5-й разряд RUS , что вызывает I запрет (запрет на прием внешних запросов) магистрали крейта.
4. CALL RJCTI (CMC(4)) - очищает 5-й разряд RUS , т.е. отменяется I запрет в магистрали крейта.
5. CALL TSTI(J)(CMC(5, J)) - где J - целая переменная. Проверяется наличие запрета в магистрали крейта. J = I, если установлен I запрет; J = 0 - в противном случае.
6. CALL SICLS (CMC(6)) - устанавливает II-й разряд RUS , что приводит к запрету генерации цикла магистрали крейта при выполнении по ОШ операций типа "чтение с паузой", адресованных к модулям КАМАК. Это ускоряет выполнение инструкции MOV , в которой в качестве приемника используется регистр КАМАК.
7. CALL RJCTS (CMC(7)) - очищается II-й разряд RUS .
8. CALL CIRCL (Z , C , I , S) (CMC(8, Z , C , I , S)) - где Z , C , I , S = 0 или I.

При единичном значении соответствующих параметров выполняются действия, указанные в пунктах 1,2,3,6.

Управление запросами и линиями запросов

9. CALL MASKS(I) (CMC(9,I)) - маскируется линия запросов RMZ (регистра масок и запросов), на которую коммутирован запрос от I-й станции (см.стр.12).
10. CALL RMASKS(I)(CMC(10, I)) - размаскировывается линия запросов RMZ , на которую коммутирован запрос от I-й станции.

В процессе работы предыдущих двух подпрограмм выдается сообщение об ошибке и выходе из программы, если была попытка маскировать или размаскировывать запрос от станции, не выдающей L - запрос. Ошибка выдается в виде: "? CAMAC-ERR-I STATION DON'T GENERATE L".

11. CALL MASKL (I) (CMC(11,I)) - маскируется I-я линия запросов RMZ .
12. CALL RMASKL (I) (CMC(12,I)) - размаскировывается I-я линия запросов RMZ .
13. CALL MASKO (CMC(13)) - маскируются все линии запросов RMZ.
14. CALL RMASKO (CMC(14)) - размаскировываются все линии запросов RMZ .
15. CALL LAM (CMC(15)) - ожидание суммарного L от не замаскированных линий запросов.
16. CALL LAMT(I) (CMC(16,I)) - где I - целая переменная. Проверяется наличие суммарного L от незамаскированных линий запросов. I = I, если есть суммарный L , I = 0 - в противном случае.
17. CALL LNA(IN,IA,IF) (CMC(17,N , I , F)) - где IN , IA , IF - целые переменные, константы или выражения. Происходит ожидание запроса L по функции IF от подстанции IA станции IN.

Управление прерываниями

18. CALL INTRPX([ISK[,IRCP]]) (CMC(18)[ск[, PCP]]) - где ISK, IRCP - целые константы, переменные или выражения. Организуется обработка прерывания по X - ответу (LX = 0). Адрес программы обработки прерывания - ISK , содержимое регистра состо-

яния процессора - ИРСП . По этим данным строится вектор прерывания. В вызове подпрограммы могут отсутствовать или ИРСП , или ИСК и ИРСП вместе. По умолчанию программа обработки прерывания определена как сообщение об ошибке X ответа на команду NAF ("? SAMAC-ERR - NO AN X - ANSWER") и выход в монитор. При этом ИРСП = 200₈.

При использовании этой подпрограммы устанавливается возврат из подпрограммы, описанных в п.21-25 без проверки 14-го разряда RUS - разряда ответа X модуля КАМАК на адресуемую команду, что несколько ускоряет выполнение этих подпрограмм.

19. CALL INTRPT(N[, ICK[, ИРСП]) (CMC(19), N[, СК[, ИРСП]]) - где N , ICK , ИРСП - целые константы, переменные или выражения. Организуется обработка прерывания по запросу от N - й станции. ICK и ИРСП - счетчик команд и регистр состояния для построения вектора прерывания. В вызове подпрограммы могут отсутствовать или ИРСП , или ИРСП и ICK вместе. По умолчанию ICK равно адресу входа, зарезервированного в конце исходного модуля библиотеки для введения по этому адресу программы обработки прерывания на языке Макроассемблер, а ИРСП = 200₈. Выдается сообщение об ошибке ("? SAMAC-ERR N STATION DON'T GENERATE L") при попытке организовать обработку прерывания по запросу от станции, не генерирующей запрос L .

В предыдущих двух подпрограммах обработка прерывания организуется следующим образом:

- а) размаскировывается соответствующая линия запросов RMZ ;
- б) размаскировывается индивидуальная маска запроса при ответе X = 0 - 10-й разряд RUS (это только для подпро-

граммы, указанной в п.18);

в) строится вектор прерывания;

г) разрешается работа схемы прерывания.

20. CALL LETI (CMC(20)) - устанавливается 6-й разряд RUS , т.е. разрешается работа схемы прерывания контроллера. Этот разряд автоматически сбрасывается после выполнения операции прерывания по ОШ для предотвращения некоторых повторных прерываний по одной и той же причине. Поэтому, чтобы обработался следующий сигнал прерывания той же причины, необходимо до его поступления (в программе обработки предыдущего прерывания или после него) вызвать эту подпрограмму.

Задание NAF команд

21. CALL NAF(IN, IA, IF) или I=NAF(IN, IA, IF) (CMC(21, N , A , F , I)), где IN , IA , IF - целые константы, переменные или выражения; IN - номер станции, IA - субадрес, IF - функция. Выполняется команда NAF , ответ Q на эту команду возвращается как результат I : I = I, если есть Q ответ; I = 0 - в противном случае.
22. CALL NAFRD(IN, IA, IF, ID) или I=NAFRD(IN, IA, IF, ID) (CMC(22, N , A , F , D , I)), где IN , IA , IF - целые константы, переменные или выражения; ID - целая переменная. Командой NAF читается с модуля КАМАК 16-разрядная информация в ID . Ответ Q возвращается как результат I : I = I, если есть Q - ответ; I = 0 - в противном случае.
23. CALL NAFWRT(IN, IA, IF, ID) или I=NAFWRT(IN, IA, IF, ID) (CMC(23, N , A , F , D , I)). Выполняется так же, как и в п.22, но только производится запись информации из ID в модуль КАМАК.

ID -целая константа, переменная или выражение.

24. CALL MODMOV(IN, IA, IN1, IA1) или I=MODMOV(IN, IA, IN1, IA1)(СМС(24, N, A, N1, A1, I)), где IN, IA, IN1, IA1 - целые константы, переменные или выражения.

Из модуля IN с субадреса IA передается 16-разрядная информация в модуль IN1 с субадресом IA1. Ответ Q возвращается как результат I: I = I, если есть Q - ответ; I = 0 - в противном случае.

25. CALL ICMS(J, IN, IA, IF[, IX[, IY]]) или I=ICMS(J, IN, IA, IF[, IX[, IY]]) (СМС(25, J, N, A, F[, X[, Y]]I)), где J=0,1,2; IX, IY - целые переменные.

Данная подпрограмма-функция выполняет следующие действия в зависимости от значения J:

а) J = 0 - чтение из модуля КАМАК по команде NAF 16-разрядного слова в IX или 24-разрядного слова, если присутствует и IY (в IY записывается старший байт);

б) J = 1 - запись в модуль КАМАК по команде NAF 16-разрядного слова из IX или 24-разрядного, если присутствует и IY (старший байт из IY);

в) J = 2 - выполняется команда NAF, а IX и IY не используются (при вызове их можно опустить).

Во всех случаях Q ответ возвращается как результат целой переменной I.

Подпрограммы-функции 21-24 предназначены для более быстрого по сравнению с 25 обмена данными между ЭВМ и модулями КАМАК, но занимают больший объем памяти.

Наряду с основными действиями, в подпрограммах-функциях, описанных выше, производится также контроль на наличие X-ответа на команду NAF, и выдается сообщение об ошибке при

его отсутствии.

Для обмена между ЭВМ и модулями КАМАК 16-разрядными словами без информации о состоянии Q и X ответов предусмотрены следующие подпрограммы-функции:

26. CALL KEEP (IN, IA, IF) (СМС(26, N, A, F)) - формируется команда NAF для его последующего использования.

27. CALL EXEC (СМС(27)) - выполняется команда NAF, зарезервированная последней из встречающейся в программе подпрограммой-функцией KEEP.

28. CALL READ (ID) (СМС(28, D)), где ID - целая переменная. Выполняется чтение 16-разрядной информации из модуля КАМАК в ID по команде NAF, зарезервированной последней подпрограммой KEEP.

29. CALL WRITE(ID) (СМС(29, D)), где ID - целая константа, переменная или выражение. Выполняется запись ID в модуль КАМАК по команде NAF, зарезервированной последней подпрограммой KEEP.

Подпрограмма KEEP должна быть вызвана до вызова подпрограмм 27-29 хотя бы один раз с соответствующими IN, IA, IF. Для организации контроля над ответом X для этих подпрограмм можно использовать подпрограмму INTRPX.

Все подпрограммы-функции реализованы на языке Макроассемблер и занимают в ОЗУ ~ 800₈ слов. Для адаптации пакета к конкретной аппаратной конфигурации в тексте на языке Макроассемблер необходимо выполнить следующие изменения:

а) значение адреса RUS,

б) базовый адрес векторов прерывания - VINT,

в) поле байтов масок - MASK следующим образом: 0-й

байт подя MASK должен содержать I в j - м разряде (в остальных 0), если запрос от внутреннего прерывания скоммутирован на j -ю линию запросов RMZ ; MASK+1 -й байт должен содержать нули во всех разрядах, если i -я станция не вырабатывает запрос L , и I в j - м разряде (в остальных 0), если запрос от i -й станции скоммутирован на j -ю линию запросов RMZ (i = 1, ..., 24; j = 1, ..., 8).

В ПРИЛОЖЕНИИ приводится текст демонстрационной программы на языке ФОРТРАН с использованием подпрограмм-функций описанного пакета, а также пакета подпрограмм [3], предназначенного для работы с графическим дисплеем, выполненным на базе модулей стандарта КАМАК.

Постановка задачи

Организуется обработка прерывания по запросу от 7-й станции, поступающего в установленные промежутки времени. Обработка этого прерывания заключается в увеличении счетчика поступившего прерывания на I. Затем принимается целое число из 6-го модуля, запоминается в массиве, и эти числа гистограммируются на графический дисплей. На графический дисплей выводится также среднее значение и дисперсия распределения поступающих чисел. Наряду с обработкой данного прерывания организуется обработка также двух других прерываний: прерывания по ответу X = 0 на команду NAF (обработка по умолчанию, см. описание подпрограммы INTRPX); прерывания по запросу от 5-й станции (обработка которой состоит в выводе сообщения END OF PROGRAM и в окончании работы всей программы).

```

INTEGER S(256)
LOGICAL*1 BS(5)
COMMON/I/ I
IT="177777
IT1="1
ISUM=500
CALL IJKLZ
CALL INTRPX
CALL ICMC(2,7,0,26)
CALL INTRPT(7,"4000","200)
CALL INTRPT(5,"4040","200)
15 AX=0
AY=0
I=0
CALL OS('ЧИСЛА','КОЛИЧЕСТВО',
*РАСПРЕДЕЛЕНИЕ ЧИСЕЛ',AX,AY,0)
DO 18 K=1,256
S(K)=0
18 S**НАЧАЛО СБОРА
14 CALL ICMC(1,7,0,16,IT,IT1)
CALL LNA(7,0,8)
CALL ICMC(0,6,0,0,K)
S(K)=S(K)+1
CALL GPOINT(20+K,6+S(K))
CALL LETI
IF(T.NF.ISUM) GO TO 14
S**КОНЕЦ СБОРА
S**ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ РАСПРЕДЕЛЕНИЯ
DO 4 K=1,256
IF(S(K).GT.0) GO TO 5
4 CONTINUE
5 K1=K
DO 6 K=1,256
IF(S(257-K).GT.0) GO TO 7
6 CONTINUE
7 K2=257-K
30 N=0
A=0
R=0
DO 8 K=K1,K2
A=A+K*S(K)
N=N+S(K)
A=A/N
DO 9 K=K1,K2
B=B+(K-A)*(K-A)*S(K)
R=R*SQRT(B/(N-1))
CALL DTEXT(45,0,'A=')
ENCODE(5,101,BS) A
CALL DTEXT(BS,5)
CALL DTEXT(45,1,'S=')
ENCODE(5,101,BS) B
CALL DTEXT(BS,5)
101 FORMAT(F5.1)
ENCODE(5,108,BS) ISUM
CALL DTEXT(45,2,'ВСЕГО=')
CALL DTEXT(BS,5)
CALL DTEXT(45,4,STR)
CALL DTEXT(' ')
108 FORMAT(I5)
PAUSE 'CONTINUE'
GO TO 15
END

```

! ИНТЕРВАЛ ВРЕМЕНИ (В МКС) ПОСТУПЛЕНИЯ СОБЫТИИ.
! КОЛИЧЕСТВО СОБИРАЕМЫХ ЭЛЕМЕНТОВ.
! ОБЩИЙ СБРОС.
! ОРГАНИЗАЦИЯ ПЕРЕРЫВАНИЯ ПО X-ОТВЕТУ.
! NAF26 ДЛЯ СБРОСА CLOCK TIMER.
! ПЕРЕРЫВАНИЕ ДЛЯ ЧТЕНИЯ ДАННЫХ.
! ПЕРЕРЫВАНИЕ ДЛЯ ОКОНЧАНИЯ РАБОТЫ.
! НА ЭКРАНЕ ГРАФИЧЕСКОГО ДИСПЛЕЯ,
! ПОДСОЕДИНЕННОГО К ЭВМ ЧЕРЕЗ МОДУЛИ
! КАМАК, ЧЕРТЯТСЯ ОСИ ПО ЗАДАНЫМ
! ПАРАМЕТРАМ (ТЕКСТОВЫЕ НАДПИСИ ОСЕЙ,
! МАСШТАБ 1:1 ПО ОБЕИМ ОСЯМ).
! ОЧИСТКА МАССИВА РАСПРЕДЕЛЕНИЯ ПОС-
! ТУПАВШИХ ЭЛЕМЕНТОВ.
! ЗАДАНИЕ ИНТЕРВАЛА ВРЕМЕНИ.
! ОЖИДАНИЕ L.
! ЧТЕНИЕ ЧИСЛА.
! МАССИВ РАСПРЕДЕЛЕНИЯ ЧИСЕЛ.
! ВЫВОД ТОЧКИ НА ЭКРАН.
! РАЗРЕШЕНИЕ СЛЕДУЮЩЕГО ПЕРЕРЫВАНИЯ.
! КОНЕЦ СБОРА?
! НАХОЖДЕНИЕ НИЖНЕЙ ГРАНИЦЫ РАС-
! ПРЕДЕЛЕНИЯ ЭЛЕМЕНТОВ.
! K1-НИЖНЯЯ ГРАНИЦА РАСПРЕДЕЛЕНИЯ.
! НАХОЖДЕНИЕ ВЕРХНЕЙ ГРАНИЦЫ РАС-
! ПРЕДЕЛЕНИЯ ЭЛЕМЕНТОВ.
! K2-ВЕРХНЯЯ ГРАНИЦА РАСПРЕДЕЛЕНИЯ.
! N-ЧИСЛО ЭЛЕМЕНТОВ В ИНТЕРВАЛЕ
! РАСПРЕДЕЛЕНИЯ, A-СРЕДНЯЯ РАСПРЕ-
! ДЕЛЕНИЯ, B-ДИСПЕРСИЯ (ПОЛУШИРИНА).
! ВЫЧИСЛЕНИЕ СРЕДНЕГО ЗНАЧЕНИЯ
! ПРИНЯТЫХ ЭЛЕМЕНТОВ.
! ВЫЧИСЛЕНИЕ ДИСПЕРСИИ (СРЕДНЕГО
! ОТКЛОНЕНИЯ) ДАННОГО РАСПРЕДЕЛЕ-
! НИЯ.
! ДАЛЕЕ ВЫВОДЯТСЯ НА ЭКРАН ГРА-
! ФИЧЕСКОГО ДИСПЛЕЯ СРЕДНЕЕ ЗНА-
! ЧЕНИЕ, ДИСПЕРСИЯ, ОБЩЕЕ КОЛИЧЕСТ-
! ВО ЭЛЕМЕНТОВ НА ИНТЕРВАЛЕ ДАН-
! НОГО РАСПРЕДЕЛЕНИЯ.
! ВСЕ ЭТО, КАК И ВЫЧЕРЧИВАНИЕ ОСЕЙ
! И ВЫСВЕЧИВАНИЕ ТОЧЕК (СМ. ВЫШЕ)
! ВЫПОЛНЯЕТСЯ ПРИ ПОМОЩИ ПОДПРОГ-
! РАММ ИЗ ДРУГОЙ БИБЛИОТЕКИ-ПРЕД-
! НАЗНАЧЕННОЙ ДЛЯ РАБОТЫ С ГРАФИ-
! ЧЕСКИМ ДИСПЛЕЕМ, ПОДСОЕДИНЕННЫМ
! К ЭВМ ЧЕРЕЗ СПЕЦИАЛЬНЫЕ МОДУЛИ
! КАМАК.
! ОЖИДАНИЕ НАЖАТИЯ (VK).
! ИДТИ НА ОЧЕРЕДНОЙ СБОР.

```

.MCALL .PRINT      *** УКАЗАТЕЛЬ СТЕКА УСТАНОВИТЬ В 1000
.ASECT            *** ДЛЯ ПРЕДОТВРАЩЕНИЯ ВОЗМОЖНЫХ ПЕРЕ-
.=42              *** КРЫТИИ РАЗНЫХ МОДУЛЕЙ ПРОГРАММЫ.
.WORD 1000        ***
;=====
.FSECT I,RW,D,GBL,REL,OVR *** ОБЩИЙ БЛОК I
.WORD            ***
;-----
.ASECT            ***** ОБРАБОТКА ЗАПРОСА L7 (Т.Е. ЗАПРОСА НА
.=4000           ***** ПОСТУПЛЕНИЕ ОЧЕРЕДНОГО ЭЛЕМЕНТА ОТ МО-
INC I            ***** ДУЛЯ КАМАК) СОСТОИТ В ВЫЧИСЛЕНИИ ПО-
RTI              ***** РЯДКОВОГО НОМЕРА ПОСТУПИВШЕГО ЭЛЕМЕНТА.
;=====
.MCALL .EXIT      *** ОБРАБОТКА ЗАПРОСА L5 (Т.Е.
.ASECT            *** ЗАПРОСА ОТ 5-ОГО МОДУЛЯ)
.=4040           *** СОСТОИТ В ТОМ, ЧТО ПЕЧАТА-
.PRINT #MSG      *** ЕТСЯ СООБЩЕНИЕ ОБ ОКОНЧАНИИ
.EXIT            *** РАБОТЫ ПРОГРАММЫ И ПЕРЕДА-
MSG: .ASCIZ / END OF PROGRAM/ *** ЕТСЯ УПРАВЛЕНИЕ МОНИТОРУ.
.EVEN           ***
.END

```

СПИСОК ЛИТЕРАТУРЫ

1. Бабаян Г.Е.,Зверев А.М. Расширение языка БЭЙСИК для программирования в стандарте КАМАК на ЭВМ типа "Электроника-100/16". Препринт ЕФИ-426(33)-80,Ереван,1980.
2. Контроллер крейта КК. Эксплуатационная документация. ПКЗ.057.000 ТО.
3. Бабаян Г.Е.,Кетикян А.Ж.,Оганисян Г.А. Программное обеспечение дисплейного набора модулей "Динамо" для работы с контроллером ККИ6. Препринт ЕФИ-799(26)-85, Ереван,1985.

Рукопись поступила 20 марта 1985 г.