

**СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА**

P10-86-92

М.Насоди, О.В.Стрекаловский

КАМАКРО - МЕТОД ПРОГРАММИРОВАНИЯ
АППАРАТУРЫ
В СТАНДАРТЕ КАМАК

1986

Для управления различными физическими установками широко применяются ЭВМ серии СМ и аппаратура в стандарте КАМАК^{7/1}.

Для управления магистралью КАМАК в ЛЯР ОИЯИ используются контроллеры К106, производства ПНР, и С1.22, производства ВНР.

Для программирования контроллеров на ЭВМ СМ3, СМ4, ТРА1140, МЕРА60/30 успешно используется специальный набор макроопределений КАМАКРО. При разработке способа программирования КАМАКРО необходимо было удовлетворить следующие требования:

- программы должны работать на машинах семейства СМ, независимо от конкретного типа машины /СМ или Электроника 60/, то есть они должны быть одинаковыми на уровне исходного языка;

- программы для различных контроллеров /К106 или С1.22/ должны быть одинаковыми на уровне исходного языка;

- скорость исполнения камаковских функций должна быть приближена к аппаратной;

- задержка ответа системы на запросы LAM должна быть минимальной,

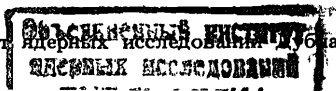
- параллельно выполняющиеся программы должны управлять различными модулями, находящимися в одном крейте, не приводя к возникновению конфликтных ситуаций.

В зависимости от типа используемого процессора для одного и того же макровывода исходного языка генерируются различные управляющие коды. Формы управления контроллером, которые нельзя реализовать контроллером другого типа, не используются. Так, не используется возможность контроллера К106 с помощью одной инструкции вызывать последовательные циклы ввода и вывода.

Регистры устройств адресуются непосредственно. Такие параметры, как номер крейта, модуля, субадреса определяются во время трансляции, и только данные циклов ввода и вывода являются динамическими параметрами. Поэтому изменение номера крейта или модуля требует повторной трансляции программы, но эта маленькая трудность окупается скоростью выполнения программы.

Получение минимальной реакции системы на запросы LAM осуществляется при использовании прерываний непосредственно. В операционной системе RSX11M прямо используется директива присоединения прерываний CINT. Внутренняя администрация задерживает прерывание не более чем на 50 мкс. В системе РТ11 надо использовать непосредственно адреса векторов прерывания.

При параллельной работе нескольких программ, когда требуется управлять модулями независимо друг от друга, обслуживание запро-



сов автоматически осуществляется отдельно по модулям. При этом в случае квазипараллельного управления модулями одного крейта нельзя разделять связанные инструкции управления, например цикл вывода, состоящий из нескольких инструкций, и проверка ответа Q. Решение проблемы осуществлено с помощью временного запрещения прерываний с помощью маскирования. В ЭВМ типа СМЗ, СМ4 для маскирования используется возможность временно устанавливать приоритет процессора равным семи, запрещая прерывания.

СПИСОК МАКРООПРЕДЕЛЕНИЙ КАМАКРО, ПРИМЕНЯЕМЫЙ ДЛЯ ПРОГРАММИРОВАНИЯ АППАРАТУРЫ В СТАНДАРТЕ КАМАК НА МАШИНАХ ТИПА СМ

Объяснение значений, используемых при описании макропараметров:

параметр	значение	пример
I	целое	7
P	выражение	mov RO,R1
M	параметр типа макро	Vary
X..Q	X или Q бит./Используется только X или только Q/.	
C	C=0, если контроллер типа C1.22, C=1, если контроллер K106.	

TYPE тип контроллера /C1.22 или K106/.

После знака = стоит значение, присваиваемое по умолчанию.

Параметр, заключенный в скобки, является необязательным и может отсутствовать.

МАКРОПАРАМЕТРЫ КАМАКРО

САМАС TYPE

- определяется, какой из наборов макроопределений, в зависимости от типа контроллера, будет использован.

NAF CRATE MODUL SUBADR FUNC (K (TESTED))

I=C I I=0 I X..Q

- выполняется функция NAF, и, если требуется, проверяется правильность выполнения. При неуспешной операции устанавливается бит переноса в слове состояния процессора. Если требуется, то при ошибке команда будет повторена K раз, если ранее не будет правильного выполнения.

AREAD CRATE MODUL SUBADR OPLOW OPHIGH FUNC (K (TESTED))

I=C I I=0 M M I=0 I X..Q

- чтение 24-разрядного слова с магистрали КАМАК. Проверяется, как для команды NAF. Данные передаются в OPLOW и OPHIGH в последовательности:

MOV ,OPLOW
MOVB ,OPHIGH

Пример: AREAD 1, 12.,, RO,R1,,3,Q

Контроллер K106, 24-разрядное слово будет прочитано в регистрах RO и R1 из модуля, занимающего 12-ю станцию. Проверяется появление разряда Q при правильном выполнении операции. Если была ошибка, то будет сделано три попытки повторить операцию.

BREAD CRATE MODUL SUBADR OPERAND FUNC (K (TESTED))

I=C I I=0 M I=0 I X..Q

- чтение 8 младших разрядов с магистрали КАМАК. Проверка, как в случае NAF. Байт передается в OPERAND по команде MOVБ.

CREAD CRATE MODUL SUBADR OPERAND FUNC (K (TESTED))

I=C I I=0 M I=0 I X..Q

- чтение 16-разрядного слова с магистрали КАМАК. Проверка, как в случае NAF. Данные передаются в OPERAND по команде MOV.

AWRITE CRATE MODUL SUBADR OPLOW OPHIGH FUNC (K (TESTED))

I=C I I=0 M M I=16 I X..Q

- запись 24-разрядного слова. Проверка, как в случае NAF.

Данные передаются из OPLOW и OPHIGH по командам:

MOVБ ,OPHIG

MOV ,OPLOW

BWRITE CRATE MODUL SUBADR OPERAND FUNC (K (TESTED))

I=C I I≠0 M I=16 I X..Q

- запись 8-разрядного слова. Проверка, как в случае NAF. Байт передается из OPERAND по инструкции MOVБ.

CWRITE CRATE MODUL SUBADR OPERAND FUNC (K (TESTED))

I=C I I=0 M I=16 I X..Q

- запись 16-разрядного слова. Проверка, как в случае NAF. Данные передаются из OPERAND по команде MOV.

CSET CRATE PARAM

I M

- устанавливает в соответствии с заданными разрядами регистра состояния контроллера. Например, C SET 1, # C.IE установит в контроллере K106 разряд разрешения прерывания.

SSET CRATE PARAM

I M

- выборочная установка разрядов в регистре состояния контроллера. Например, SSET 0, # C.ONL!C.IE установит в регистре состояния контроллера C1.22 разряды ON-LINE и разрешение прерывания.

SCLEAR CRATE PARAM

I M

- выборочное обнуление разрядов в регистре состояния контроллера. Например, SCLEAR 0, # C.XER обнулит разряд ошибки в C1.22.

STEST CRATE PARAM

I M

- проверка установленных разрядов в регистре состояния контроллера в соответствии со значением PARAM. Используется форма # C.X.!C.Q.

ONL CRATE (K)
I I

- установка контроллера в выбранном крейте в состоянии готовности. Проверка, как в случае NAF.

SETINT CRATE MODUL
I I

- разрешение обработки прерывания, вызванного сигналом LAM, выставленным соответствующим модулем. Обязательно для контроллера K106.

VECTOR VARI CRATE MODUL
M I I

-макропеременная VARI во время трансляции получает значение адреса вектора прерывания, связанного со значением сигнала LAM данного модуля. После этого VARI можно использовать как адрес вектора прерывания.

При написании программы следует обращать внимание на используемые переменные. Ниже приведен список переменных, которые можно использовать, но нельзя переопределять:

C.CL C.Z. C.IE C.X. C.Q C.ONL C.XER C.MC CNA MASKK
LENABLE ; F26
LDESAB ; F24
LCLEAR ; F10
LTEST ; F8

МАКРООПРЕДЕЛЕНИЯ, РАЗРАБОТАННЫЕ ДЛЯ ЕДИНООБРАЗНОГО ПРОГРАММИРОВАНИЯ НА АССЕМБЛЕРЕ ДЛЯ РАЗЛИЧНЫХ СМ ЭВМ

Значения, используемые для описания макропараметров:

параметр	значение	пример
I	целое	7
P	параметр типа инструкции	MOV RO,R1
S	инструкция, в которой отсутствует параметр выхода	BIS 5
M	параметр,используемый в инструкциях	VARY
C	условие,которое может возникнуть после символа B в инструкциях условных ветвлений	C=CS BCS
TYPE	тип процессора	P11.40,SM4,SM3

После знака = стоит значение, присваиваемое по умолчанию.
PLM1 TYPE
TYPE

- выбирает и определяет макроопределения в зависимости от типа процессора. Тип процессора может быть выбран из ряда: SM4, SM3, LSI11, P11.40.

LET A (B) (S...S)
M M S S

- выполняется последовательность инструкций, которая изменяет значение первого параметра. Например, LET R3 R2 (BIC # 4) NEG означает MOV R2, R3; BIC # 4,R3; NEG R3

LETB A (B) (S...S)
M M S S

- аналогично описанной выше последовательности инструкций, но первая команда MOVB.

CLEAR (A...A)
M M

- переменные обнуляются инструкцией CLR. Например,CLEAR RO, (SP) означает CLR RO; CLR (SP)

PUSH (A...A)
M M

MOV A,-(%6)

- значения переменных сохраняются в стеке. Например,PUSH RO,R2 означает MOV RO,-(%6); MOV R2,-(%6)

POP (A...A)
M M

MOV (%6) ,A

- переменным присваиваются значения, находящиеся в стеке.

DO (A...A)
P P

- выполняется последовательность инструкций в списке. Например, DO (CALL RUT) (BCS ERROR)

GOTO L

- передача управления к метке L.

GOTO L (C)
L R

- передача управления к метке L, если выполняется условие R. Например, GOTO L CS означает BCS L.

OTOG L (C)

- передача управления к метке L, если условие не выполняется.

WHILE A B C
M C M

- оператор цикла с предусловием. Пока выражение A, управляющее повторением, удовлетворяет условию B, выполняется C.

Например: WHILE (TSTB(RO)) NE (MOVB (RO) ,R1)

транслируется в:
1\$: TSTB (RO)
BEQ 2\$
MOVB (RO) ,R1
BR 1\$

2S:

UNTIL A B C
M C M

- оператор цикла с постусловием. Оператор C будет выполняться до тех пор, пока управляющее повторением выражение A не будет B. Например: UNTIL (CMPB(RO) ,R1) EQ (INC R2)

транслируется в: 1\$: CMPB (RO)+,R1
 BEQ 2\$
 INC R2
 BR 1\$
 2\$:

ELIHW A B C
 - оператор цикла, аналогичный WHILE. Однако проверка условия повторения делается после выполнения тела цикла C. Например:
 ELIHW (CMP RO,R1) LE ADD R2,R3

транслируется в: 1\$: ADD R2,R3
 CMP RO,R1
 BLE 1\$

LITNU A B C
 M C M
 - аналогично UNTIL, однако проверка B происходит после выполнения тела цикла C. Например: LITNU (TST RO) MI (CLR R3)

1\$: CLR R3
 TST RO
 BMI 1\$

CYCLE (A) B C
 R P M
 - быстрый цикл. Тело цикла C будет повторено B раз. A - регистр, используемый для переменной цикла. Например:

CYCLE RO,R1,MOVB (R3)+,(R4)+
 транслируется в: MOV R1,RO
 1\$: MOVB (R3)+,(R4)+
 SOB RO,1\$

если регистр для переменной цикла не задан, то CYCLE,R1,CLR-(RO)
 транслируется в: MOV R1,2\$
 1\$: CLR -(RO)
 DEC +(R7)
 2\$: .WORD 0
 BNE 1\$

IF A B C THEN D (ELSE E)
 P C P M M
 - условный оператор. Если при сравнении выражений A и C результат будет равен B, то выполняется выражение D. В противном случае E. Например: IF RO LT 'A' THEN LET RO 'B'

CMP RO, 'A'
 BLT 1\$
 MOV 'B',RO

1\$:
 Если в операторе условия используется GOTO, то при выполнении условия ветвления осуществляется переход к метке L. Например: IF RO HIS # 1986, GOTO PASS
 транслируется в: CMP RO, # 1986.
 BHIS PASS

В операторе условия может быть только сложная команда и проверяемое условие, то есть IF A B GOTO C.
 Например: IF CS (QIO\$\$ PAR...) GOTO ERROR

QIO\$\$ PAR...
 BCS ERROR
 SHL A B SHR A B
 I P I P

-арифметические сдвиги A раз влево и A раз вправо параметра B.

При разработке программного обеспечения, связанного с управлением аппаратурой в стандарте КАМАК, применяется следующий метод. Основная программа пишется на языке высокого уровня Паскаль или Фортран. Когда требуется работа с аппаратурой, вызываются подпрограммы, написанные на ассемблере с использованием набора макроопределений "КАМАКРО". Приведен пример, в котором при появлении сигнала ЛАМ, сигнализирующего о готовности информации в регистре модуля, происходит чтение этой информации и вывод ее на экран дисплея.

PROGRAM PRIMER;
 FUNCTION INILINE: BOOLEAN; EXTERNAL;
 FUNCTION GETWORD: INTEGER; EXTERNAL;
 BEGIN

IF INILINE THEN
 REPEAT
 Writeln (GETWORD)
 UNTIL FALSE
 ELSE Writeln ("ERROR")
 END.

В основной программе, написанной на языке Паскаль, используются внешние функции, написанные с использованием КАМАКРО.

.MCALL DECL\$\$,SETF\$\$,CLEF\$\$,WTSE\$\$,ASTX\$\$,CINT\$\$
 BASE:
 INBUF: .WORD 0 ; буфер ввода информации
 ..M.. = 1 ; отметка кода, вызываемого с высоким уровнем
 CISR: ; уровнем, код прерывания
 CALL LDESA ; запрет выставления ЛАМ
 SSET CRATE,# C.IE; разрешение прерывания для крейта
 CALL @ # \$FORK2; снятие высокого приоритета
 ..M..=0; конец кода, выполняемого с высоким уровнем
 CLR @ R3
 IF CS (CREAD CRATE,MO21,,INBUF,,0,Q) GOTO ENPRI; ,если
 ; была ошибка, тогда переход
 MOV X.TCB-X.FORK(R5),R5; установить флаг
 MOV #1,RO ; отметка прекращения ожидания
 CALL @ # \$SETF
 FERR: RETURN
 ENPRI: NAF CRATE,MO21,,LENABL; разрешение блоку ставить ЛАМ
 RETURN
 LDESA:

..M..=1 ; эта часть кода вызывается с высоким уровнем
NAF CRATE,MO21,,LDESAB; запрет блоку ставить LAM
RETURN

..M..=0 ; конец кода, выполняемого с высоким уровнем
; функция, для получения 16-разрядного слова с блока
FUNC GETWORD PACK,INTEGER,0

BEGIN

CLEF\$\$ # 1 ; очистить флаг

CALL ENPRI ; разрешить прерывания

WTSE\$\$ # 1 ; ожидание ввода

MOV INBUF,PACK(SP); присвоить функции полученное

ENDPR ; значение

; логическая функция для инициализации и проверки

FUNC INILINE WORD,BOOLEAN,0

PARAM GEP,1

SAVE (RO)

BEGIN

CLR WORD(SP); установка значения "ложь"-ошибка

IF CS (ONL CRATE,2) GOTO NOSUCC; ошибка в крейте?

IF CS (CINT\$\$ V21S,BASE,CISR,LDESA,PR7) GOTO NOSUCC

SETINT CRATE, MO21; разрешение прерываний общее

INC WORD(SP) ; установка значения "истина"

NOSUCC: ENDPR

.END

Используемый в примере модуль различает следующие функции:
FO - чтение информации, F26 и F24 - разрешение и запрещение
для блока устанавливать сигнал требования обслуживания LAM.

Функция INILINE служит для подготовки контроллера в выбранном крейте для работы с ЭВМ, разрешения прерываний и установки логического соответствия между конкретным физическим модулем и вектором прерывания. Функция GETWORD предназначена для чтения информации с блока КАМАК при выставлении им запроса обслуживания LAM. Использование прерываний приводит к тому, что программа не занимает постоянно время центрального процессора, освобождая его для решения других задач.

При трансляции используются различные библиотеки. Если программа в приведенном выше примере состоит из файлов PRIMER.PAS и CAMIN.MAC, то команды трансляции следующие:

MAC CAMIN =LB:(1,1)CAMAC/ML,PASMAC,SY:(7,3)CAMIN

PAS PRIMER=PRIMER

При построении задачи из полученных объектных модулей необходимо учесть следующие требования. В системах с диспетчером памяти следует присоединять к задаче область регистров внешних устройств DEVCOM, и, если используются прерывания, объявить задачу привилегированной. Пример построения задачи PRIMER.TSK:

TKB PRIMER/CP/PR:0=PRIMER,CAMIN,LB:(1,1)PASILIB/LB

/
COMMON= DEVCOM:RW
//

Набор макрорасширений КАМАКРО используется в Лаборатории ядерных реакций с 1981 года. Опыт применения метода КАМАКРО при написании программного обеспечения для физической установки "СИЗИФ"^{2/}, драйвера для управления графопостроителем, разработки графического пакета управления телевизионными мониторами, управления модулями в локальной сети ЭВМ ЛЯР^{3/} и целого ряда других программ показал, что, во-первых, сокращается время, требуемое на разработку программ, во-вторых, получаемый текст легче для сопровождения, внесения изменений и дополнений, яснее выделена логика работы программы, в-третьих, широкое использование режима прерываний приводит к экономному использованию центрального процессора, в-четвертых, значительно облегчился перенос программного продукта между СМ ЭВМ с различными типами процессоров и контроллеров КАМАК. При необходимости использовать отличные от K106 и C1.22 контроллеры или СМ ЭВМ с более широким набором команд надо к уже существующему набору добавить соответствующие макрорасширения. Система КАМАКРО является открытой для внесения изменений и дальнейшего расширения.

ЛИТЕРАТУРА

1. CAMAC a modular Instrumentation System for Data Handling. ESONE Committee. EUR 4100, 1972.
2. NIM 84-220-2,3 p.419.
3. Б.В.Фефилов и др. В сб.: XI Международный симпозиум по ядерной электронике /Братислава 1983/ ОИЯИ, Д13-84-53, Дубна, 1984.

Рукопись поступила в издательский отдел
17 февраля 1986 года.

Насоди М., Стрекаловский О.В.

P10-86-92

КАМАКРО - метод программирования аппаратуры
в стандарте КАМАК

Описывается метод программирования КАМАКРО, разработанный и используемый в Лаборатории ядерных реакций ОИЯИ для управления аппаратурой в стандарте КАМАК. Работает с операционными системами RT11 и RSX11M. Исходный текст программы, написанный с использованием КАМАКРО, является одинаковым для разных типов контроллеров и процессоров СМ ЭВМ. Конкретный вид кодирования определяется при трансляции. Решен вопрос управления модулями, находящимися в одном крейте, при многопрограммном, многопользовательском режиме работы. Система "КАМАКРО" является открытой для дальнейшего расширения.

Работа выполнена в Лаборатории ядерных реакций ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1986

Перевод О.С.Виноградовой.

Nasodi M., Strekalovsky O.V.

P10-86-92

CAMACRO - A Method for Programming CAMAC Apparatus

"CAMACRO" - a new method for programming CAMAC systems is described. This method can be used on computer controlled by RT11 or RSX11M operation systems. Source code is the same for different crate controllers and PDP11-type central processors. Concrete assembler code is defined at the translation stage. The method is intended for controlling CAMAC modules in one crate in multiprogramming, multiusers' regime. The "CAMACRO" system is open for further development.

The investigation has been performed at the Laboratory of Nuclear Reactions, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1986