

Рис. 5.

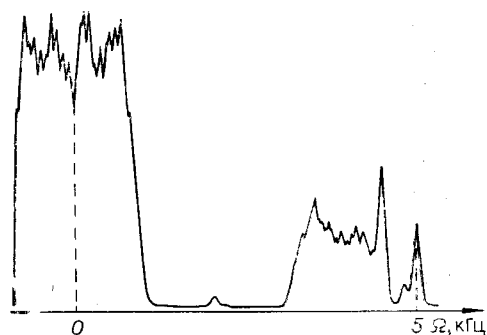


Рис. 6.

ляющаяся критерием качества демодулированного сигнала, $V = 2400$ бит/с для канала протяженностью 4 переприемных участка.

На рис. 4 приведены осциллограммы для 8-уровневого способа передачи, т. е. $V_{\text{такт}} = 7200$ бит/с, на рис. 5 — спектр модулированного сигнала, на рис. 6 — спектр демодулированного сигнала (до фильтра низких частот). Конструктивно модем выполнен в виде модуля КАМАК тройной ширины и предназначен для работы совместно с модулями «Передатчик данных» и «Приемник данных», реализующими процедуры блочной высокоуровневой передачи данных (ISO-3309) в системах передачи данных.

ЛИТЕРАТУРА

1. Бобров Е. Н., Слепнев В. А., Фесенко Б. В. Модули КАМАК, ориентированные на создание терминальных комплексов различного назначения.— *Автометрия*, 1980, № 4.
2. Курицын С. А., Перфильев Э. П., Пономарев В. И. Формирование спектра сигнала при передаче данных.— *Электросвязь*, 1975, № 12.

Поступила в редакцию 27 июля 1979 г.

УДК 621.3.06

К. П. ВЛАХОВА

(София, Болгария)

ЯЗЫК ПРОМЕЖУТОЧНОГО УРОВНЯ ДЛЯ УПРАВЛЕНИЯ МИКРОПРОЦЕССОРНОЙ СИСТЕМОЙ КАМАК

1. Введение. Язык IML предназначен для работы с системами КАМАК в режиме реального времени. Семантика языка определена комитетом по стандартизации в области ядерной электроники (ESONE) [1], а синтаксис связан с конкретной реализацией [2]. В настоящей работе описан синтаксис языка промежуточного уровня для управления системами КАМАК с помощью микропроцессора серии 6800.

2. **Операторы описания.** Эти операторы задают имя и атрибуты определенной величины в программе, а также резервируют память в зависимости от атрибутов.

2.1. Операторы описания для КАМАК-адресов. Эти операторы резервируют два байта памяти для каждого КАМАК-адреса.

2.1.1. Описание КАМАК-адресной константы.

LOCD<h-имя>,H,<вспа>
 <h-имя>:: = <идентификатор>
 <идентификатор>:: = <буква>{<буква>|<цифра>}₀⁵
 <буква>:: = A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
 <цифра>:: = 0|1|2|3|4|5|6|7|8|9
 <вспа>:: = <OC>(,<c>,<n>,<a>) (<c>,<n>,<a>)<ЗС>
 <OC>:: = (
 <ЗС>:: =)
 в этой версии не употребляется
 <c>:: = 0|1 номер крейта
 <n>:: = 0|1|...|31 номер модуля
 <a>:: = 0|1|...|15 номер регистра (субадрес)

2.1.2. Описание КАМАК-адресной переменной.

LOCD<p-имя>,P|,<вспа>|

Если <вспа> опущено, то переменная не имеет определенного начального значения.

2.1.3. Описание массива заданных КАМАК-адресов.

LOCD <g-имя>, G, <#-вспа> {[,<вспа>]}₀⁹⁶
 <g-имя>:: = <идентификатор>
 <#-вспа>:: = <десятичное число>
 <десятичное число>:: = <цифра>[<десятичное число>]

Этот оператор генерирует <#-вспа> КАМАК-адресов по следующим правилам:

- если <#-вспа> меньше количества заданных адресов <вспа>, то генерируются только первые <#-вспа> адресов;
- если <#-вспа> больше количества заданных адресов <вспа>, то для недостающих адресов резервируется память;
- если список <вспа> пуст, то для массива только резервируется память.

2.1.4. Описание массива вычисляемых КАМАК-адресов.

LOCD<c-имя>,C,<#-вспа>[,<вспа-Н>,<вспа-к>,<вспа-с>]
 <c-имя>:: = <идентификатор>
 <вспа-Н>:: = <вспа> адрес начала
 <вспа-к>:: = <вспа> адрес конца
 <вспа-с>:: = <вспа> шаг

Этот оператор генерирует <#-вспа> КАМАК-адресов следующим образом:

- если <#-вспа> больше количества вычисляемых адресов, то массив дополняется нулями;
- если <#-вспа> меньше количества вычисляемых адресов, то генерируются лишь <#-вспа> адресов;
- если <вспа-Н>,<вспа-к>,<вспа-с> опущены, то для массива резервируется только память.

2.1.5. Описание сканирующей группы.

LOCD<q-имя>,Q,<вспа-Н>,<вспа-к>
 <q-имя>:: = <идентификатор>

Этот оператор последовательно генерирует два КАМАК-адреса.

2.1.6. Описание массива заданных КАМАК-адресов с элементами, доступными по индексу.

LOCD <gs-имя>, GS, <#-вспа> {[,<вспа>]}₀⁹⁶
 <gs-имя>:: = <идентификатор>

Массив этого типа отличается от массива G-типа по способу до-

ступа к элементам. Массив используется только полностью, тогда как у GS-массива доступен каждый элемент.

2.1.7. Описание массива вычисляемых КАМАК-адресов с элементами, доступными по индексу.

LOCD<cs-имя>, CS, <#-bсна>[,<bсна-Н>,<bсна-к>,<bсна-с>]
<cs-имя>:: = <идентификатор>

Массив типа CS отличается от массива типа C способом доступа к элементам; в массиве CS доступен каждый элемент.

2.2. Операторы описания для памяти. Эти операторы резервируют один или три байта памяти каждого элемента в зависимости от длины слова.

2.2.1. Описание константы.

LOCM<k-имя>, K, <длина слова>, <самоопределенный терм>
<k-имя>:: = <идентификатор>
<длина слова>:: = 1|3
<самоопределенный терм>:: = <десятичное число>|B' {<двоичная цифра>}₁²⁴|X' {<шестнадцатеричная цифра>}₁⁶
<двоичная цифра>:: = 0|1
<шестнадцатеричная цифра>:: = <цифра> A|B|C|D|E|F

При помощи этого оператора резервируется один или три байта памяти в зависимости от длины слова. Значение константы определяется самоопределенным термом.

2.2.2. Описание переменной.

LOCM<v-имя>, V, <длина слова>
<v-имя>:: = <идентификатор>

Переменная занимает один или три байта.

2.2.3. Описание логической переменной.

LOCM<vl-имя>, VL
<vl-имя>:: = <идентификатор>

Логическая переменная занимает один байт памяти и в процессе исполнения программы принимает значение 0 или 1.

2.2.4. Описание массива.

LOCM<a-имя>, A, <длина слова>, <длина массива>
<a-имя>:: = <идентификатор>
<длина массива>:: = <число>

Этот оператор задает массив элементов, число которых равно параметру <длина массива>, а длина каждого элемента (в байтах) определяется параметром <длина слова>.

2.2.5. Описание массива с элементами, доступными по индексу.

LOCM<as-имя>, AS, <длина слова>, <длина массива>
<as-имя>:: = <идентификатор>

Массив этого типа отличается от массива А-типа по способу доступа к элементам.

2.3. Описание запросов. 2.3.1. Описание запроса, доступного по субадресу (LAM-A).

LOCL<l-имя>, SUBI, [<приоритет>], <адрес прерывания>
<l-имя>:: = <идентификатор>
<приоритет>:: = 0|1...|15
<адрес прерывания>:: = <идентификатор>|<самоопределенный терм>

Этот оператор генерирует в двух байтах <bсна>. Если указан приоритет, то его значение заносится в таблицу приоритетов, если указан адрес прерывания, то его значение заносится в вектор прерываний.

2.3.2. Описание запроса, доступного по бит-позиции в регистре (LAM-1).

LOCL<l-имя>, BIT, <бит-позиция>, <bсна>[, [<приоритет>], <адрес прерывания>]
<бит-позиция>:: = <самоопределенный терм> со значением, равным 2^k , где $k = (0,23)$.

$\langle \text{bcp} \rangle ::= \langle \text{OC} \rangle (\langle \text{b} \rangle, \langle \text{c} \rangle, \langle \text{n} \rangle) (\langle \text{c} \rangle, \langle \text{n} \rangle) (\langle \text{c} \rangle, \langle \text{n} \rangle) \langle \text{ЗС} \rangle$

Оператор генерирует в двух байтах $\langle \text{bcp} \rangle$, в трех байтах — $\langle \text{бит-позиция} \rangle$. Для этого оператора все изложенное выше о параметрах $\langle \text{приоритет} \rangle$ и $\langle \text{адрес прерывания} \rangle$ остается в силе.

2.4. Операторы описания для системы прерываний. 2.4.1. Описание таблицы приоритетов.

$\text{PRTY} \langle \text{OC} \rangle \{ \langle \text{приоритет} \rangle \} \{ \{ \langle \text{приоритет} \rangle \} \}_0^{31} \langle \text{ЗС} \rangle$

В зависимости от позиции в списке этот оператор генерирует соответствующий $\langle \text{приоритет} \rangle$ в таблице приоритетов.

2.4.2. Описание вектора прерываний.

$\text{IRQV} \langle \text{c} \rangle, \langle \text{OC} \rangle \{ \langle \text{адрес прерывания} \rangle \} \{ \{ \langle \text{адрес прерывания} \rangle \} \}_0^{31} \langle \text{ЗС} \rangle$

Оператор генерирует $\langle \text{адрес прерывания} \rangle$ в соответствующем векторе прерываний в зависимости от $\langle \text{c} \rangle$.

3. Операторы действия. Эти операторы определяют действия, реализуемые в КАМАК-программе с помощью одной КАМАК-команды или цепочки КАМАК-команд. Операторы действия подразделяются на три класса в зависимости от метода доступа к модулям.

3.1. Операторы простого действия. При использовании любого F-кода выполняется один КАМАК-цикл.

3.1.1. Безусловный оператор простого действия.

1) Чтение:

$\{ \langle \text{метка} \rangle \} \text{SA} \langle \text{f-чтение} \rangle, \langle \text{КАМАК-1} \rangle, \langle \text{память-1} \rangle \{ \langle \text{конец} \rangle \}$
 $\langle \text{f-чтение} \rangle ::= \text{F00} | \text{RD1} | \text{F01} | \text{RD2} | \text{F02} | \text{RC1} | \text{F03} | \text{RCM} | \text{F04} | \text{F05} | \text{F06} | \text{F07}$
 $\langle \text{КАМАК-1} \rangle ::= \langle \text{h-имя} \rangle | \langle \text{p-имя} \rangle | \langle \text{bcp} \rangle | \langle \text{КАМАК-s} \rangle$
 $\langle \text{КАМАК-s} \rangle ::= \langle \text{OC} \rangle (\langle \text{gs-имя} \rangle, \langle \text{индекс} \rangle) (\langle \text{cs-имя} \rangle, \langle \text{индекс} \rangle) \langle \text{ЗС} \rangle$
 $\langle \text{индекс} \rangle ::= \langle \text{k-имя} \rangle | \langle \text{v-имя} \rangle | \# \langle \text{самоопределенный терм} \rangle$
 $\langle \text{память-1} \rangle ::= \langle \text{v-имя} \rangle | \langle \text{память-s} \rangle$
 $\langle \text{память-s} \rangle ::= \langle \text{OC} \rangle \langle \text{as-имя} \rangle, \langle \text{индекс} \rangle \langle \text{ЗС} \rangle$
 $\langle \text{конец} \rangle ::= \langle \text{буква} \rangle (\text{индикация для последней инструкции в КАМАК-программе}).$

2) Запись:

$\{ \langle \text{метка} \rangle \} \text{SA} \langle \text{f-запись} \rangle, \langle \text{КАМАК-1} \rangle, \langle \text{память-2} \rangle \{ \langle \text{конец} \rangle \}$
 $\langle \text{f-запись} \rangle ::= \text{F16} | \text{WT1} | \text{F17} | \text{WT2} | \text{F18} | \text{SS1} | \text{F19} | \text{SS2} | \text{F20} | \text{F21} | \text{SC1} | \text{F22} | \text{F23} | \text{SC2}$
 $\langle \text{память-2} \rangle ::= \langle \text{k-имя} \rangle | \langle \text{память-1} \rangle$

3) Операция, не использующая шины данных:

$\{ \langle \text{метка} \rangle \} \text{SA} \langle \text{f-операция} \rangle, \langle \text{КАМАК-1} \rangle \{ \langle \text{конец} \rangle \}$
 $\langle \text{f-операция} \rangle ::= \text{F08} | \text{TLM} | \text{F09} | \text{CL1} | \text{F10} | \text{CLM} | \text{F11} | \text{CL2} | \text{F12} | \text{F13} | \text{F14} | \text{F15} | \text{F24} | \text{DIS} | \text{F25} | \text{XEQ} | \text{F26} | \text{ENB} | \text{F27} | \text{TST}$

4) Чтение статуса:

$\{ \langle \text{метка} \rangle \} \text{SA} \langle \text{f-статус} \rangle, \langle \text{КАМАК-1} \rangle, \langle \text{v1-имя} \rangle \{ \langle \text{конец} \rangle \}$
 $\langle \text{f-статус} \rangle ::= \text{RSTAT} | \text{RLAM}$

3.1.2. Оператор простого действия с переходом по $Q = 1$. Если после выполнения КАМАК-цикла $Q = 1$, то управление передается по адресу перехода; если $Q = 0$, выполняется следующий оператор в программе.

1) Чтение:

$\{ \langle \text{метка} \rangle \} \text{SJQ} \langle \text{f-чтение} \rangle, \langle \text{КАМАК-1} \rangle, \langle \text{память-1} \rangle, \langle \text{адрес перехода} \rangle, \langle \text{конец} \rangle$
 $\langle \text{адрес перехода} \rangle ::= \langle \text{идентификатор} \rangle | \langle \text{самоопределенный терм} \rangle$

2) Запись:

$\{ \langle \text{метка} \rangle \} \text{SJQ} \langle \text{f-запись} \rangle, \langle \text{КАМАК-1} \rangle, \langle \text{память-2} \rangle, \langle \text{адрес перехода} \rangle, \langle \text{конец} \rangle$

3) Операция, не использующая шины данных:

$\{ \langle \text{метка} \rangle \} \text{SJQ} \langle \text{f-операция} \rangle, \langle \text{КАМАК-1} \rangle, \langle \text{адрес перехода} \rangle, \langle \text{конец} \rangle$

3.1.3. Оператор простого действия с переходом по $Q = 0$. Если после выполнения КАМАК-цикла $Q = 0$, то управление передается по

адресу перехода, а если $Q = 1$, выполняется следующий оператор.

1) Чтение:

[<метка>]SJNQ<f-чтение>, <КАМАК-1>, <память-1>, <адрес перехода>, <конец>

2) Запись:

[<метка>]SJNQ<f-запись>, <КАМАК-1>, <память-2>, <адрес перехода>, <конец>

3. Операция, не использующая шины данных:

[<метка>]SJNQ<f-операция>, <КАМАК-1>, <адрес перехода>, <конец>

3.2. Операторы передачи блока данных. С помощью этих операторов осуществляется передача блоков данных между КАМАК-регистрами и памятью ЭВМ.

3.2.1. Передача блока данных с синхронизацией по запросу (LAM). Модуль устанавливает LAM, когда он готов выполнить очередную операцию. Передача блока заканчивается, когда содержимое счетчика становится равным нулю.

1) Чтение:

[<метка>]UBL<f-чтение>, <КАМАК-1>, <память-3>, <LAM-1>, <счетчик>, [<конец>]

<память-3> ::= <a-имя>|<as-имя>|<память-s>

<LAM-1> ::= <l-имя>

<счетчик> ::= <v-имя>|<k-имя>|#<самоопределенный терм>

где значение параметра <самоопределенный терм> должно быть меньше 256.

2) Запись:

[<метка>]UBL<f-запись>, <КАМАК-1>, <память-3>, <LAM-1>, <счетчик>, [<конец>]

3.2.2. Передача блока данных. Передача блока заканчивается, когда содержимое счетчика становится равным нулю.

1) Чтение:

[<метка>]UBC<f-чтение>, <КАМАК-1>, <память-3>, <счетчик> [<конец>]

2. Запись:

[<метка>]UBC<f-запись>, <КАМАК-1>, <память-3>, <счетчик> [<конец>]

3.2.3. Передача блока данных с синхронизацией по $Q = 0$. Передача прекращается, если $Q = 1$ или содержимое счетчика равно 0.

1) Чтение:

[<метка>]UBR<f-чтение>, <КАМАК-1>, <память-3>, <счетчик> [<конец>]

2) Запись:

[<метка>]UBR<f-запись>, <КАМАК-1>, <память-3>, <счетчик> [<конец>]

3.2.4. Передача блока данных с синхронизацией по $Q = 1$.

1) Чтение:

[<метка>]UBS<f-чтение>, <КАМАК-1>, <память-3>, <счетчик> [<конец>]

2) Запись:

[<метка>]UBS<f-запись>, <КАМАК-1>, <память-3>, <счетчик> [<конец>]

3.3. Операторы мультимодульного действия. Эти операторы осуществляют передачу блока данных между несколькими модулями или выполняют одну операцию с несколькими модулями. Операция окончена, когда перебраны все модули.

3.3.1. Оператор мультимодульного действия.

1) Чтение:

[<метка>]MA<f-чтение>, <КАМАК-2>, <память-3>, <счетчик>, <конец>
<КАМАК-2> ::= <g-имя>|<c-имя>|<gs-имя>|<cs-имя>

2) Запись:

[<метка>]MA<f-запись>, <КАМАК-2>, <память-3>, <счетчик>, <конец>

3) Операция, не использующая шины данных:

[<метка>]MA<f-операция>, <КАМАК-2>, <счетчик> [<конец>]

3.3.2. Оператор мультимодульного действия с синхронизацией по $Q = 0$.

1) Тест LAM:

[<метка>]MNQ TLM, <КАМАК-2> [,<конец>]

2) Тест статуса:

[<метка>] MNQ TST, <КАМАК-2>, [,<конец>]

3.3.3. Оператор мультиадресного сканирования.

1) Чтение:

[<метка>]MAD<f-чтение>, <КАМАК-3>, <память-3>, [,<конец>]

2) Запись:

[<метка>]MAD<f-запись>, <КАМАК-3>, <память-3> [,<конец>]

3) Операция, не использующая шины данных:

[<метка>]MAD<f-операция>, <КАМАК-3>, <память-3> [,<конец>]

3.3.4. Выбор номера станции.

1) Запись:

[<метка>]SNR<f-запись>, <КАМАК-1>, <память-1>, <номер станции> [,<конец>]

<номер станции> ::= <v-имя>|<k-имя>|#<самоопределенный терм>

2) Операция, не использующая шины данных:

[<метка>]SNR<f-операция>, <КАМАК-1>, <номер станции> [,<конец>]

4. Операторы для работы с запросами.

1) Оператор разрешения запроса:

[<метка>]ENL<LAM-1> [,<конец>]

Этим оператором осуществляется разрешение запроса, доступного по субадресу (LAM-A), с помощью команды F(26) либо разрешение запроса, доступного по бит-позиции (LAM-1), с помощью команды F(19) A(13).

2) Оператор запрещения запроса:

[<метка>]DISL<LAM-1>, [,<конец>]

Оператор осуществляет запрещение LAM-A с помощью команды F(24) либо запрещение LAM-1 с помощью команды F(23) A(13).

3) Оператор сброса запроса:

[<метка>]CLRL<LAM-1> [,<конец>]

Оператор осуществляет сброс запроса по команде F(10) для LAM-A и F(23) A(12) для LAM-1.

4) Проверка наличия запроса и переход:

[<метка>]IFL<LAM-1>, <адрес перехода>, [,<конец>]

Осуществляется проверка запроса с помощью команды F(8) для LAM-A или проверка по бит-позиции в регистре с субадресом A(14) для LAM-1. При наличии запроса выполняется переход по заданному адресу.

5) Проверка отсутствия запроса и переход:

[<метка>]RLR<bcn>, <память-1'> [,<конец>]

Отличается от предыдущего оператора тем, что переход по заданному адресу выполняется при отсутствии запроса.

6) Проверка состояния и переход при наличии запроса:

[<метка>]IFS<LAM-1>, <адрес перехода> [,<конец>]

Осуществляется проверка состояния источника запроса с помощью команды F(27) для LAM-A или проверка по бит-позиции для LAM-1. При наличии запроса выполняется переход по заданному адресу.

7) Проверка состояния и переход при отсутствии запроса:

[<метка>]IFNS<LAM-1>, <адрес перехода> [,<конец>]

Отличается от предыдущего оператора тем, что переход по заданному адресу выполняется при отсутствии запроса.

8) Чтение:

[<метка>]RL<LAM-1>, <память-1'> [,<конец>]

<память-1'> ::= <v-имя>

Осуществляется проверка запроса с помощью команды F(8) для LAM-A или проверка по бит-позиции с последующей записью в память.

9) Чтение регистра состояний источников запроса:

[<метка>]RLS <всп>, <память-1'> [,<конец>]

Осуществляется перенос информации в память по команде F(1) A(12).

10) Чтение регистра состояний требований:

[<метка>]RLR<всп>, <память-1'> [,<конец>]

Этот оператор выполняет передачу содержимого регистра A(14) в память при помощи команды F(1).

11) Чтение регистра маски запросов:

[<метка>]RLM <всп>, <память-1'> [,<конец>]

Содержимое регистра A(13) передается в память по команде F(1).

12) Запись регистра маски запросов:

[<метка>]WLM <всп>, <память-2'> [,<конец>]

<память-2'> ::= <к-имя>|<память-1'>

Оператор заносит информацию из памяти в регистр A(13).

5. Операторы общего управления. Эти операторы предоставляют доступ к крейт-контроллеру.

1) Оператор запуска:

[<метка>]CZ<с>

С помощью оператора запуска генерируется сигнал на шине Z.

2) Оператор сброса:

[<метка>]CC<с>

С помощью оператора сброса генерируется сигнал на шине C.

3) Оператор запрета:

[<метка>]SETCI<с>

Генерируется сигнал на шине I.

4) Оператор снятия запрета:

[<метка>]CLRCI<с>

5) Оператор разрешения запроса от крейта:

[<метка>]ENCD<с>

6) Оператор запрещения запроса от крейта:

[<метка>]DISCD<с>

7) Оператор перехода по запрету:

[<метка>]IFCI<с>, <адрес перехода>

Если I = 1, то управление передается на метку <адрес перехода>.

8) Оператор перехода при отсутствии запрета:

[<метка>]IFNCI<с>, <адрес перехода>

Если I = 0, то управление передается на <адрес перехода>.

9) Оператор перехода по разрешению запроса от крейта:

[<метка>]IFCD<с>, <адрес перехода>

10) Оператор перехода по запрещению запроса от крейта:

[<метка>]IFNCD<с>, <адрес перехода>

11) Оператор чтения слова состояния крейт-контроллера:

[<метка>]RCS<с>, <память-1'>

При помощи этого оператора слово состояния крейт-контроллера переносится в память.

6. Операторы для работы с системой прерываний. Эти операторы дают возможность передачи служебных приоритетов в таблицу и вектор прерываний.

1) Изменение приоритета и адреса прерывания:

[<метка>]CHGL<всп>I,<приоритет>]I,<адрес прерывания>]

Оператор загружает приоритет и адрес прерывания; если один из этих параметров опущен, то соответствующее действие не выполняется.

2) Изменение граничного приоритета и адреса прерывания:

[<метка>]LMTP<с>I,<приоритет>]I,<адрес прерывания>]

3) Изменение приоритета и адреса прерывания, связанных с возникновением ошибки:

[<метка>]XERI<с>I,<приоритет>]I,<адрес прерывания>]

4) Изменение приоритета и адреса прерывания, связанных со вторым счетчиком времени:

[<метка>]TMOF<с>[,<приоритет>][,<адрес прерывания>]

5) Изменение приоритета и адреса прерывания, связанных с третьим счетчиком времени:

[<метка>]TMOS<с>[,<приоритет>][,<адрес прерывания>]

6) Изменение приоритета и адреса прерывания, связанных с часами:

[<метка>]TIMER<с>[,<приоритет>][,<адрес прерывания>]

7) Изменение приоритета и адреса прерывания, связанных с арифметико-логическим устройством:

[<метка>]ALUI<с>[,<приоритет>][,<адрес прерывания>]

8) Изменение приоритета и адреса прерывания, связанных с периферийным интерфейсным адаптером:

[<метка>]PIAI<с>[,<приоритет>][,<адрес прерывания>]

9) Изменение приоритета и адреса прерывания, связанных с индикацией конца КАМАК-программы:

[<метка>]EOCR<с>[,<приоритет>][,<адрес прерывания>]

7. **Заключение.** Язык IML обеспечивает доступ к аппаратно-реализуемым операциям в режиме реального времени. Синтаксис связан с системой АС-02.

ЛИТЕРАТУРА

1. The Definition of IML.— ESONE Committee. 1974.
2. Kubitz M., Kind R. Macro-IML Manual. Berlin, НМI-B162, 1975.

Поступила в редакцию 6 декабря 1979 г.

УДК 681.3

**А. Н. КАСПЕРОВИЧ, Т. Н. МАНТУШ, В. И. ПРОКОПЕНКО,
В. А. СЛУЕВ, В. И. СОЛОНЕНКО**
(Новосибирск)

СИСТЕМА ДЛЯ МИКРОПРОГРАММИРОВАНИЯ И ОТЛАДКИ УСТРОЙСТВ НА СЕКЦИОННЫХ МИКРОПРОЦЕССОРАХ

В настоящее время для построения быстродействующих устройств вычислительной техники широко применяются секционные микропроцессоры (СП) [1, 2]. Обычно в состав набора БИС СП входят БИС центрального процессорного элемента (ЦПЭ), БИС блока микропрограммного управления (БМУ) и др. Добавляя к этому набору БИС ПЗУ и стандартные схемы малой и средней степеней интеграции, можно создавать разнообразные вычислительные устройства. Блок микропрограммного управления совместно с ПЗУ в таких устройствах выполняет управляющие функции, а совокупность ЦПЭ — арифметические и логические операции.

Создание устройств на базе СП усложняется (по сравнению с ситуацией, когда применяются однокристалльные микропроцессоры) тем, что в данном случае приходится решать задачи не только программирова-