

54270 б.

В-676

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

756/10741
ЭКЗ. Ч. 10741

11 - 5427 +



А.И. Волков

АВТОКОД MADLEN
(ОПИСАНИЕ ТРАНСЛЯТОРА)

ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
И АВТОМАТИЗАЦИИ

1970

1 - 5427

А.И. Волков

АВТОКОД MADLEN
(ОПИСАНИЕ ТРАНСЛЯТОРА)

ОБЩЕСТВЕННАЯ БИБЛИОТЕКА
ЦЕНТРАЛЬНАЯ БИБЛИОТЕКА
ИМЕНИ МОТОВА

В в е д е н и е

В системе программирования ДУБНА /4/ автокод *MADLEN* применяется в случаях, когда при программировании необходимо максимально учитывать особенности машины БЭСМ-6, а также в качестве промежуточного языка при трансляции с ФОРТРАНа. В будущем, возможно, автокод *MADLEN* используют трансляторы с каких-либо других алгоритмических языков. Таким образом, эффективность трансляции в системе существенно зависит от скорости трансляции с автокода. Это обстоятельство повлияло на структуру транслятора с автокода, предусматривающего различные варианты работы в зависимости от "типа" пользователя.

Так как транслятор с автокода написан на автокоде *MADLEN* и текст транслятора снабжен подробными комментариями, данное описание не ориентировано на подробное изложение всех деталей. В сообщении описаны лишь принципиальные вопросы построения транслятора и приводятся некоторые сведения, упрощающие непосредственное ознакомление с текстом транслятора. В приложениях, кроме того, приведена инструкция к подпрограмме транслятора *SIMFAIL* *, предназначенной для работы с последовательностями (лентами) текстовых восьмиразрядных символов. Эта подпрограмма может оказаться полезной для применений, не связанных с рассматриваемым транслятором.

Предполагается, что читатель имеет общее представление о структуре мониторной системы ДУБНА, знаком с описанием автокода *MADLEN*.

I. Обращение к транслятору

Обращение к транслятору имеет вид:

- 1, *VTM* , *READ/*
- 2, *VTM* , *SAVE/*
- 3, *VTM* , *MADLEN/*
- 4, *VTM* , 4096
- 5, *VTM* , *TØ*
- , *CALL* , *MADLEN*

$T\emptyset$ - максимальная длина таблицы описаний (и поля, отведенного для ее формирования). В настоящем варианте

$$T\emptyset = 1024.$$

READ/ - подпрограмма выдачи очередной инструкции транслируемой подпрограммы. Инструкция всегда содержит 14 слов.

READ/ удовлетворяет библиотечным правилам и имеет один формальный параметр - адрес размещения текста инструкции.

SAVE/ - подпрограмма запоминания и восстановления состояния *READ/*. Используется для организации двух просмотров текста исходной подпрограммы. Единственный параметр задается значением сумматора перед обращением к *SAVE/*:

$\Sigma = 0$ - запоминание

$\Sigma \neq 0$ - восстановление.

Возможны две формы задания инструкции (см. 2):

а) текстовая карта (84 символа $IS\emptyset$),

б) набор понятий (слов)

$MADLEN/ = \begin{cases} MADLEN* & - \text{ работа с текстовыми картами} \\ MADLEN** & - \text{ работа с наборами понятий} \end{cases}$

$MADLEN*$ и $MADLEN**$ - подпрограммы автокода

2. Две формы инструкций

I. Текстовые карты. Текстовой картой называется последовательность символов

$$S_1 S_2 \dots S_{84},$$

где

$$S_i (i=1, \dots, 83)$$

символы $IS\emptyset/3/$, а $S_{84}=012_8$ - признак конца последовательности.

Текстовые карты являются общеупотребительным способом обмена текстами между подпрограммами монитора, к которым следует отнести и трансляторы.

Текстовые карты используются при трансляции подпрограмм непосредственно с автокода (активизируемого в этом случае монитором). В качестве фактических параметров, соответствующих формальным параметрам *READ/* и *SAVE/*, используются наименования подпрограмм постоянной библиотеки.

READ * И SAVE *
(MADLEN / соответствует MADLEN *)

2. Набор понятий. В случае, когда программа на автокоде является результатом работы транслятора с ФОРТРАНА, нет необходимости формировать обычный текст подпрограммы, требующий трудоемкой работы по выделению основных синтаксических понятий, так как таковые уже были определены на уровне трансляции с ФОРТРАНА.

Набор понятий, характеризующий одну инструкцию автокода, определяется группой из семи слов:

I. 48 47 46 45

π_c	π_I	π_L	π_D	
---------	---------	---------	---------	--

$\pi_c = I$ Содержимое набора игнорируется. На печать (в режиме *FULL LIST*) выдается строка
C*** НАЧАЛО_ОПЕРАТОРА_ N**,
 где *N* - количество наборов с $\pi_c = I$, поступившее транслятору (с учетом данного).

$\pi_I = I$ Первая составляющая адреса (предполагаемая всегда положительной) есть идентификатор.

$\pi_L = I$ Первая составляющая адреса - адрес типа "ЛИТЕРАЛ", причем соответствующая константа состоит из одного слова, если $\pi_D = 0$.

(π_{I+1}) $\pi_D = I$ Константа, **соответствующая** конструкции "адрес типа литерал", состоит из двух слов.

2.

метка в коде T E X T

3.

48		4	I
π_m		<i>MØD</i>	

$\pi_m = I$ - Инструкция помечена.

MØD - определяет значение указателя регистра в инструкции.

4.

	8 I
	<i>NM</i>

NM - порядковый номер мнемкода в таблице *KØPNAME*

5.

	A _I
--	----------------

6.

	A _I '
--	------------------

Последние два слова используются для представления первой составляющей адреса:

а) Если $\mathcal{K}_1 = I$ или мнемокод есть *VLØCK* (или *СØNT*) в A_I помещается идентификатор в коде T E X T. A_I игнорируется.

б) Если $\mathcal{K}_2 = I$ ($\mathcal{K}_1 = 0$), то соответствующая константа помещается в A_I (при $\mathcal{K}_D = 0$) или же в A_I и A_I' ($\mathcal{K}_D = I$)

в) Если мнемокод есть Ø C T (другие константы недопустимы), то константа помещается в A_I.

7.

	48	I5	I
K ₂		A2	

A2 - вторая составляющая (00000+77777₈)

$\left. \begin{array}{l} \mathcal{K}_2 = 0 + \\ \mathcal{K}_2 = I - \end{array} \right\}$ знак перед второй составляющей

Для *VLØCK* - вторая составляющая есть длина фрагмента (в адрес представим только один фрагмент). Состояние \mathcal{K}_2 безразлично.

Нетрудно заметить, что в данном случае на синтаксис автокода накладывается ряд ограничений: инструкция *VLØCK* имеет только один фрагмент в адресе, значение указателя регистра не может быть идентификатором и т.п. При необходимости эти ограничения могут быть сняты посредством увеличения понятий в наборе. Данный вариант набора определен спецификой трансляции с языка ФОРТРАН в системе ДУБНА.

Обращение к транслятору в этом случае осуществляется заключительным блоком транслятора с ФОРТРАНА (*GLINK*) /I/ с параметрами *READF*, *SAVEF* и *MADLEN***, соответствующими формальным параметрам *READ/*, *SAVE/* и *MADLEN/*.

3. Состав транслятора

Приведенная ниже **схема иллюстрирует** взаимодействие подпрограмм автокода и некоторых других подпрограмм (из постоянной библиотеки), используемых при трансляции с автокода:

ASSEMBLER - заполняет признак "трансляция непосредственно с ав-

токада" (*MADLEN**). Выполняет обращение к транслятору с параметрами

*READ** , *SAVE** , *MADLEN** .

Обращение к этой подпрограмме осуществляется из монитора посредством

CALL, ASEMBLER

- MADLEN* - начальная подпрограмма транслятора. Выполняет подготовку к трансляции.
- MADLEN** - блок трансляции непосредственно с автокода. Транслируемая подпрограмма - последовательность текстовых карт.
- MADLEN*** - блок трансляции с автокода. Транслируемая подпрограмма - последовательность наборов понятий

*READ**
*SAVE**
READF
SAVEF } описаны выше.

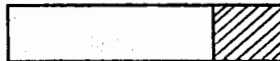
KOPNAME - таблица допустимых в *MADLEN* мнемонических кодов.

*PRIMADL** - выполняет сервисные работы при трансляции (печатать, редактирование и т.п.).

PER6B81C } вспомогательные подпрограммы для *PRIMADL* .
*SIMFAIL** }

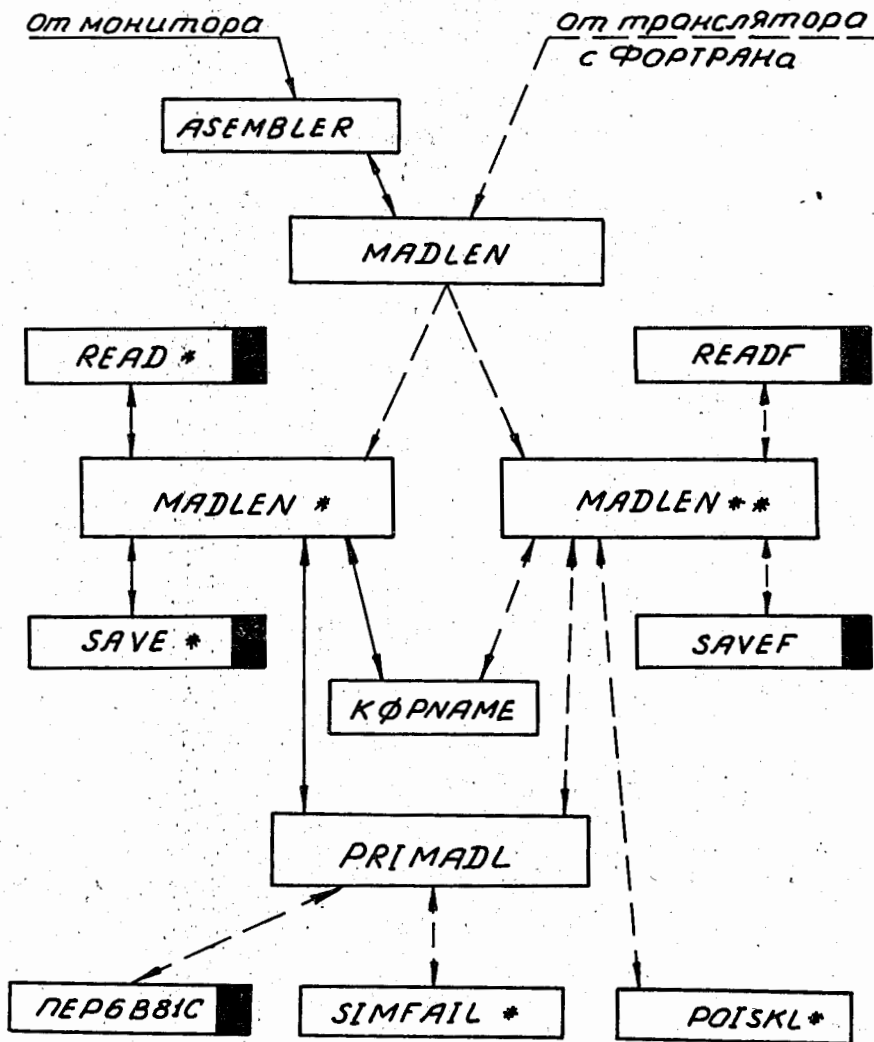
*POISKL** - часть *MADLEN*** . Выделена в отдельную подпрограмму для использования ее и транслятором с ФОРТРАНа (блок *GLINK*) .

Пунктирными линиями на схеме связаны подпрограммы, работающие только при трансляции с ФОРТРАНа. Таким образом, различным вариантам использования транслятора с автокода соответствуют различные варианты загрузки его подпрограмм. Обозначение блока



использовано для подпрограмм, не относящихся непосредственно к комплексу транслятора с автокода.

Схема I. Взаимодействие подпрограмм при трансляции с автокода



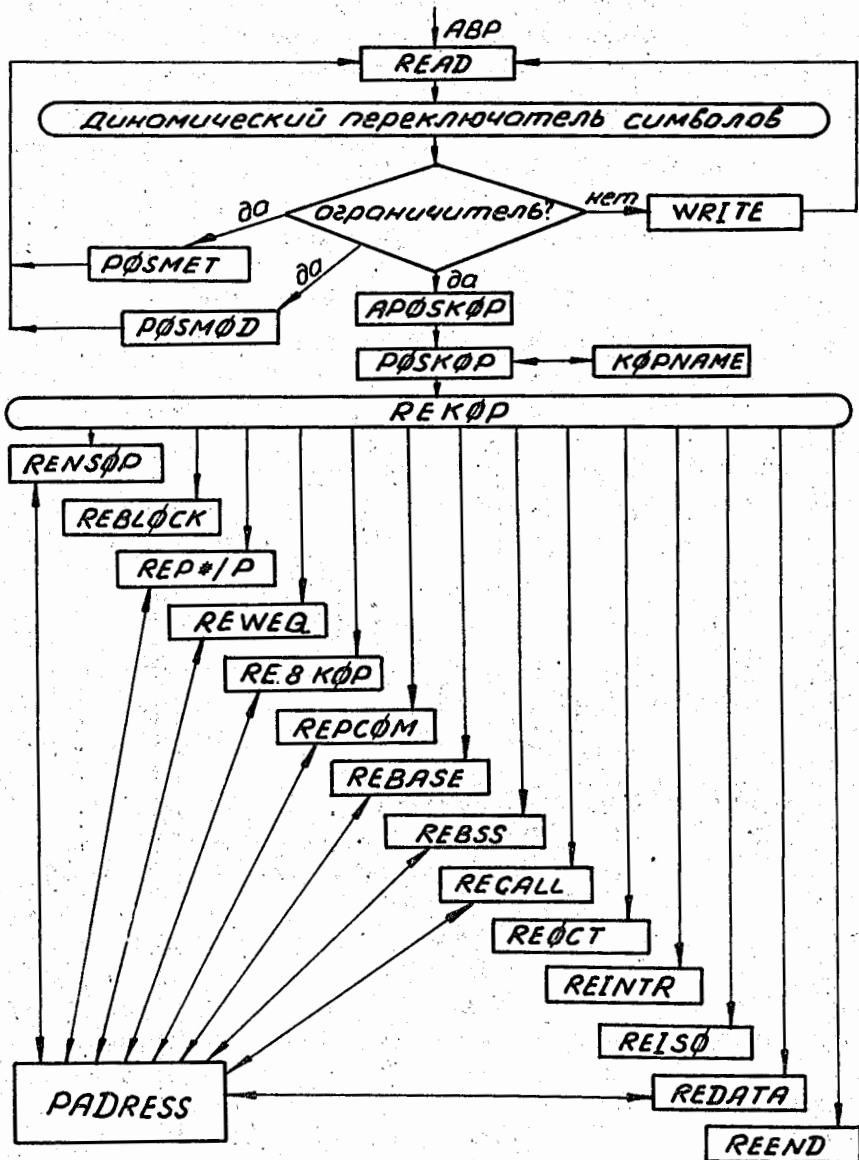
4. Схема трансляции

Трансляция осуществляется прямым (или несинтаксическим способом). При этом **проводится два просмотра текста** транслируемой подпрограммы. На первом просмотре определяются некоторые количественные характеристики (структура) стандартного массива, выполняется полная обработка меток, некоторых эквивалентностей, а также ряд операций вспомогательного характера. В результате работы транслятора на первом просмотре формируется справочная таблица, содержащая сведения о характере всех объектов, использованных в транслируемой подпрограмме (массивы, константы типа "литерал", подпрограммы и т.п.). Эта таблица размещена в массиве **SEKTOP**. Будем ее далее называть условно **таблицей идентификаторов**. На втором просмотре окончательно формируются все элементы стандартного массива и выполняется, если необходимо, печать и перфорация результатов трансляции.

Схема 2 иллюстрирует работу транслятора при выполнении одного просмотра блоком трансляции **MADLEN**. Одним из основных функциональных элементов блока является динамический переключатель символов, параметр которого есть код очередного символа транслируемой подпрограммы, поступающего от процедуры **READ**. Совокупность символов, расположенных в исходной подпрограмме между двумя последовательными ограничителями, запоминается процедурой **WRITE** и в зависимости от типов окаймляющих ограничителей и состояния переключателей передается одной из подпрограмм **P0SMET**, **P0SM0D** или **AP0SK0P**, завершающих, соответственно, обработку указателя метки, указателя **регистра** или мнемонического кода операции.

Если мнемокод не является заголовком подпрограммы (этот случай обрабатывается самой подпрограммой **AP0SK0P**), то он передается подпрограмме **P0SK0P** для окончательной обработки. С помощью таблицы мнемокодов здесь определяется параметр переключателя **REK0P**, выполняющего передачи управления на процедуры обработки конкретных типов инструкций. Эти процедуры, в свою очередь, могут потребовать выделения понятия из адресной части инструкции. Для большинства типов инструкций адрес имеет форму полного адреса, обработкой которого занимается **PADRESS**. Для других инструкций рассмотрение адреса выполняется непосредственно процедурами обработки конкрет-

Схема 2. Работа блока трансляции MADLEN *



ной инструкции. Однако почти во всех случаях выделение понятий из адресной печати выполняется с помощью уже описанного аппарата (динамического переключателя символов и процедур *READ* и *WRITE*).

Схема работы блока трансляции *MADLEN*** аналогична описанной выше. Однако эта подпрограмма не содержит элементов, связанных с выделением понятий из исходного текста. *MADLEN*** не имеет динамического переключателя, процедур *READ* и *WRITE*. В серию других процедур также внесены соответствующие исправления, обусловленные отсутствием работы по выделению понятий, а также упрощениями в синтаксисе входного языка.

5. Используемые массивы

Транслятор использует следующие массивы:

а) для связи с монитором:

- FULISFL** - признак режима *FULL LIST* ;
- NOISFL** - признак режима *NO LIST* ;
- MADLEN** - признак трансляции непосредственно с автокода;
- ENTRY** - таблица входов в подпрограмму;
- FLGEX** - характеристика ошибок трансляции.

б) собственные массивы транслятора:

- RABINF* (200) - константы и некоторые рабочие массивы;
- SECTOR* (4096) - таблица идентификаторов;
- TABDEC* (1024) - таблица описаний;
- ERRORBUF* (13) - рабочий массив процедур;
- SPASAN** (210) - рабочий массив, используемый процедурами *REISO* (*MADLEN**), *EDITOR* и *EDITORF* (*PRIMADL**);
- BUFFER* (256) - буфер для записи элементов стандартного массива во временную библиотеку.

RABINF

Список всех констант, помещенных в массив *RABINF*, приведен в подпрограмме *MADLEN*. В приложении I дан перечень наиболее часто употребляемых элементов этого массива (рабочих ячеек) с указанием специфики их использования. Поэтому ниже приводится лишь описание двух элементов этого массива (*MISTABLE* и *FLAGTABL*), используемых для управления трансляцией.

MISTABL - таблица локальных признаков трансляции. Такие признаки сохраняются, как правило, лишь на период трансляции одной инструкции.

48	47	46	45	44	43		24	23	22		I6	
----	----	----	----	----	----	--	----	----	----	--	----	--

Здесь

- 48 p. - признак ошибки в инструкции;
- 47 p. - признак ошибки в метке;
- 46 p. - признак ошибки в указателе регистра;
- 45 p. - признак ошибки в мнемокоде;
- 44 p. - признак ошибки в адресе;
- 24 p. - признак помеченности инструкции;
- 23 p. - признак команды с длинным адресом;
- 22 p. - признак "модификатор-идентификатор";
- I6 p. - признак "недопустим нулевой модификатор".

FLAGTABL - таблица глобальных признаков трансляции.

		46		43		24	23	22		5		I
--	--	----	--	----	--	----	----	----	--	---	--	---

- 46 p. - признак групп *DATA* и *SET* ;
- 43 p. - признак группы *DATA* ;
- 24 p. - признак помеченности следующей команды;
- 23 p. - признак переноса помеченности по требованию предыдущей команды;
- 5 p. - признак трансляции с автокода *MADLEN* (если 0, то трансляция с *SIBESM -6*);
- I p. - признак первого просмотра текста транслируемой подпрограммы.

SECTOR

Таблица идентификаторов содержит сведения обо всех идентификаторах (за исключением дополнительных входов) и константах, определяемых конструкциями "адрес типа литерал". Каждому такому объекту соответствуют два слова:

- 1) идентификатор в коде *TEXT* или константа,
- 2) характеристика.

Характеристика для идентификаторов имеет следующее строение:

	46	45	44	43		29...I9	I8, I7	I6		A
--	----	----	----	----	--	---------	--------	----	--	---

- 45 p. - признак неопределенности описания;
 44 p. - признак двойственного описания;
 $\checkmark = I$ - метка;
 $\checkmark = 3$ - подпрограмма;
 $\checkmark = 3,46$ p. = I. - подпрограмма, имеющая описатель ;
 $\checkmark = 4+7$ и \checkmark определяют массивы так же, как в таблице описаний;
 A - значение идентификатора;
 I6 p. - признак условного A;
 $\checkmark = 0$ - эквивалентность;
 43 p. = 0 - истинному адресу или метке в зависимости от значения I6 p.
 43 p. = I - относительному адресу A
 для констант "литерал" блока MADLEN* :

48....	46					I5			I
	\checkmark							A	

- $\checkmark = I$ - константа типа LOG ;
 $\checkmark = 2$ - тип INT (указатель I);
 $\checkmark = 3$ - тип REAL (указатель R);
 $\checkmark = 4$ - тип ISØ (указатель H);
 A - соответствующий относительный адрес.

для констант "литерал" блока MADLEN**:

48,47		39		25		I5			I
	\checkmark		A'					A	

- $\checkmark = I$ - константа из одного слова ($A^I = 0$);
 $\checkmark = 2$ - первое слово двойной константы:
 A^I - адрес в таблице второго слова;
 $\checkmark = 3$ - второе слово двойной константы ($A = A^I = 0$);
 A - относительный адрес, соответствующий константе.

6. Некоторые детали

а) использование регистров. К числу глобальных индексных регистров отнесены регистры 1,3 - 7. Основное их назначение определяется следующим списком:

- 1 - начальный адрес динамического переключателя;
- 3 и 4 - для работы процедуры *READ (MADLEN*)* ;
- 5 - для работы процедуры *WRITE (MADLEN)* ;
- 6 - *STRØKA* ;
- 7 - база констант процедуры *PADDRESS (MADLEN*)*.

б) Функция расстановки. Для работы с таблицей идентификаторов применяется аппарат функций расстановки, описанный в /2/. Применяется следующий вариант функции расстановки: идентификатор (или константа) режется на куски длиной по II разрядов, которые складываются по $\text{mod } 2^{II}$. Обозначим результат через *s*. Тогда функция расстановки определяется формулой

$$f(\text{<идентификатор>}) = A + s * 2,$$

где *A* - начальный адрес функции расстановки.

в) Подпрограмма SIMFAIL*. Эта подпрограмма приводится в приложении 2 и может быть использована для различных текстовых преобразований. Так как при создании подпрограммы основное внимание уделялось ее универсальности, в ряде случаев ее применение может привести к написанию недостаточно быстродействующих программ. Так, в подпрограмме *MADLEN** процедуры для выделения понятий *READ* и *WRITE* используют более эффективный частный алгоритм.

г) Трансляция с автокода SIBESM-6 /5/. Совместимость с автокодом *SIBESM-6* ставилась лишь как частная задача по сохранению без значительных изменений программ, написанных на этом языке для версии транслитора, используемого в течение 1969 года. Для возможных последующих версий такой совместимости может и не быть. Что же касается "совместимого" варианта *SIBESM-6*, то для его трансляции требуются следующие изменения: инструкции *DATA*, *SET*, *END* должны быть написаны в соответствии с синтаксисом автокода *MADLEN*.

д) Характеристики транслятора:

Объем транслятора при трансляции непосредственно с автокода	- ≈ 4000 слов;
Объем транслятора при использовании набора понятий	- ≈ 2500 слов;
Объем рабочих полей	- ≈ 3000 слов;
Максимальное количество объектов в подпрограмме (идентификаторов и констант "литерал")	- 2048 ;
Скорость трансляции непосредственно с автокода	- около 200 инструкций в секунду;
Скорость трансляции при использовании наборов понятий	- около 400 инструкций в секунду.

ПРИЛОЖЕНИЕ I Массив *RABINF*

RABINF : *LC*, *BLØCK*, (10), *STRØKA* (190)

- (IO) - вспомогательное поле для организации печати очередной инструкции (для хранения инструкции отведено 14 первых слов массива *STRØKA*);
- AA - начало таблицы идентификаторов;
- AAA - начало таблицы описаний;
- ABS CNT* - количество абсолютных констант;
- A ENTR* - начало таблицы входов;
- BB - конец таблицы идентификаторов;
- BBB - конец таблицы описаний;
- BASEM ØD* - значение указателя регистра в инструкции;
- BASE ** - значение (- БАЗА) при заданном автоматическом базировании;
- BASIR* - номер регистра, по которому должно производиться автоматическое базирование;
- BASPR* - признак "инструкция-команда";
- B ENTR* - конец таблицы входов;
- BSS CNT* - количество последних *BSS* ;

- DATCNT* - количество констант *DATA* ;
FLAGTABL - таблица общих признаков трансляции (см. 5);
INDTABL - рабочее поле процедуры *WRITE (MADLEN*)*
IMIST - признак ошибки при выделении понятия (или его синтаксическом анализе);
K0SCNT - количество констант "литерал";
LEFINST - ячейка для формирования очередного слова из двух команд;
MESADR1 - первая составляющая полного адреса;
MESADR2 - вторая составляющая полного адреса;
MESADR3 - значение полного адреса в случае истинности;
MESKOP - мнемокод;
MESMET - метка;
MESMOD - указатель регистра;
*MESREAD** - фактический параметр для *READ/* ;
*MESSAVE** - фактический параметр для *SAVE/* ;
MISTABL - таблица локальных признаков трансляции (см. 5);
NKABS - начальный адрес группы абсолютных констант;
NKCARD - порядковый номер карты подпрограммы;
NK0S - начальный адрес группы констант "литерал";
OUTMADLE - возврат из автокода;
P0SIND - понятие, выделенное процедурой *WRITE* ;
RMIST - признак ошибки при переводе понятий из одной системы представления в другую.

ПРИЛОЖЕНИЕ 2. Подпрограмма для работы с лентами символов *SIMFAIL**

Назовем лентой символов последовательность

$$S_1, S_2, \dots, S_N,$$

где S_i - восьмиразрядные символы (000+377₈), а N - кратно 6-ти. Предполагается, что такая последовательность плотно закодирована (в порядке следования символов) в массив ячеек оперативной памяти длиной $\frac{N}{6}$. Поставим в соответствие такой ленте указатель j , значение которого является индексом интересующего нас символа S_j

$$1 \leq j \leq N.$$

Для такой ленты определим следующие операции:

- I. Запись символа в ленту (для установленного к этому моменту значения указателя).

2. Чтение символа с ленты.
3. Увеличение указателя на единицу (операция определена для $j \leq N-1$).
4. Уменьшение указателя на единицу (операция определена для $j \geq 2$).

Так как программа предназначена для работы с несколькими лентами, то, в общем случае, ленту определяют следующие параметры:

A - начальный адрес ленты,

j - указатель ленты,

$n = \frac{N}{6}$ - количество слов, отведенных в памяти для ленты.

Для хранения параметров каждой ленты отводится характеристика X - массив из четырех слов.

Подпрограмма SIMFAIL* является набором процедур, реализующих различные операции с лентой. Эти процедуры могут быть описаны следующим образом:

```

SIMFAIL * : , SUBP,
SIMFAIL * : I, ВЛОСК, DØ * ЛЕНТ, ПЛЮС1
              , СØNT, МИНУС1, ЗАП * П1
              , СØNT, ЧТЕ * П1, П1 * ЗАП
              , СØNT, П1 * ЧТЕ, М1 * ЗАП
              , СØNT, М1 * ЧТЕ, Л1 * ТØ * Л2
              , СØNT, ЧТЕНИЕ, ЗАПИСЬ
              , СØNT, Л2 * ТØ * Л1

```

Тогда, например, обращение к процедуре ЧТЕНИЕ осуществляется следующим образом:

13, VJM, ЧТЕНИЕ

DØ * ЛЕНТ процедура для оформления характеристики перед началом работы с лентой ($j=1$)

Входные параметры

I8 - X

I9 - A

I10 - n

Здесь и ниже обозначение I_K применяется для указания содержимого индексного регистра с номером K .

$L1 * T0 * L2$

перепись характеристики $X1$ в массив $X2$

$I8 - X1,$

$I9 - X2$

$L2 * T0 * L1$

перепись характеристики $X2$ в массив $X1$:

$I8 - X1$

$I9 - X2$

Все другие процедуры реализуют операции для ленты, характеристика которой задана в $I8$.

Элементарные операции:

ЧТЕНИЕ - чтение символа. Символ помещается в $I14$.

ЗАПИСЬ - запись символа. Исходный символ должен быть помещен в $I14$.

ПЛЮС 1 - увеличение указателя на единицу. Если в результате $j = N + 1$, то $I11 \neq 0$.

МИНУС 1 - уменьшение указателя на единицу. Если в результате $j = 0$, то $I11 \neq 0$.

Все остальные процедуры реализуют композиции тех или иных элементарных операций. Так, например, инструкция

$I3, VJM, ЗАП * П1$

эквивалентна следующей паре инструкций

$I3, VJM, ЗАПИСЬ$

$I3, VJM, ПЛЮС 1$

Все подпрограммы сохраняют значения $I1 + I8$. Из остальных регистров каждая процедура гарантирует сохранение лишь тех, которые используются для задания информации к ней.

ЛИТЕРАТУРА

1. Бродчински З., Веретенев В.Ю. и др. Транслятор с языка ФОРТРАН для системы математического обеспечения БЭСМ-6. Труды Первой всесоюзной конференции по программированию, Киев, 1968.
2. Бабецкий Г.М., Бежанова М.М. и др. АЛФА-система автоматизации программирования, под редакцией Ершова. Новосибирск, 1965.
3. Волков А.И. Автокод *MADLEN*, препринт ОИЯИ, В4-11-46-54, Дубна, 1969.
4. Н.Н.Говорун, В.Ю.Веретенев, А.И.Волков, Н.С.Заикин, И.Н.Силин, Р.Н.Федорова, В.П.Шириков. Мониторная система ДУБНА для ЭВМ БЭСМ-6. Труды второй всесоюзной конференции по программированию, Новосибирск, 1970.
5. Н.П.Трифонов. Язык символического кодирования *SIBESM-6* (Методическое пособие для программирования), МГУ, М, 1967.

Рукопись поступила в издательский отдел

26 октября 1970 года.

СОДЕРЖАНИЕ

	стр
В в е д е н и е	3
1. Обращение к транслятору	3
2. Две формы инструкций	4
3. Состав транслятора	6
4. Схема трансляции	9
5. Используемые массивы	II
6. Некоторые детали	14
Приложение I	
Массив <i>RABINF</i>	15
Приложение 2	
Подпрограмма для работы с лентами символов	16
Литература	19