

Ц 8406

М-137

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

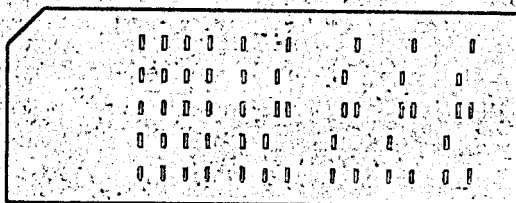


ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
И АВТОМАТИЗАЦИИ

Г.Л. Мазный

ТРАНСЛЯТОР С ЯЗЫКА МАШИНЫ БЭСМ-6
НА АВТОКОД

1970

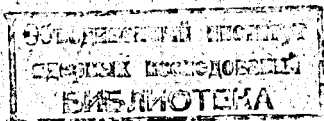


ОБЪЕДИНЕННЫЙ ИНСТИТУТ
ЯДЕРНЫХ ИССЛЕДОВАНИЙ
АВТА

11- 4950

Г.Л. Мазный

ТРАНСЛЯТОР С ЯЗЫКА МАШИНЫ БЭСМ-6
НА АВТОКОД



8282/2 48

ОГЛАВЛЕНИЕ

Введение	3
I. Общая организация перевода с машинного кода в автокод.	4
I.1. Постановка задачи.	4
I.2. Общая организация детрансляции.	6
I.3. Редактирование.	8
II. Возможности подпрограммы <i>TOMADL</i> и входная информация к ней.	10
2.1. Минимальная информация для работы <i>TOMADL</i> . Команды.	10
2.2. Автоматическая генерация и насильственное задание идентификаторов.	12
2.3. Константы и массивы рабочих ячеек типа <i>BSSN</i> .	14
2.4. Специальные массивы.	15
2.5. Общие массивы и общие блоки.	17
2.6. Эквивалентности.	18
2.7. Вызовы внешних подпрограмм и функций с помощью оператора <i>CALL</i> .	19
2.8. Входы (<i>ENTRY</i>) в детранслируемую подпрограмму.	20
2.9. Заголовок детранслируемой подпрограммы или функции.	20
2.10. Комментарии к детранслируемому объекту.	21
Литература.	22

ВВЕДЕНИЕ

В работе описывается задача "обратной трансляции" (детрансляции), которая позволяет автоматизировать процесс перевода программ из машинного кода в автокод.

Программа была написана автором в ноябре 1969 года. В настоящее время она находится в стадии опытной эксплуатации, испытывается на переводе диспетчера БЭСМ-6 Д-68-М в автокод MADLEN. Перевод диспетчера является самостоятельной крупной задачей, решение которой позволит не только получить диспетчер, к которому привязана мониторная система "Дубна", в удобочитаемом виде, но и разработать удобный аппарат внесения изменений в диспетчер с перетрансляцией его с автокода.

Для прочтения этой работы необходимо быть хорошо знакомым с /I/.

При ее написании, как и при написании самой подпрограммы TØMADL, автор пользовался постоянной поддержкой и советами В.П.Ширикова, а также принял к сведению ценные рекомендации В.Ю.Веретеннова, А.И.Волкова и И.Н.Силина.

Всем указанным лицам автор выражает свою сердечную благодарность.

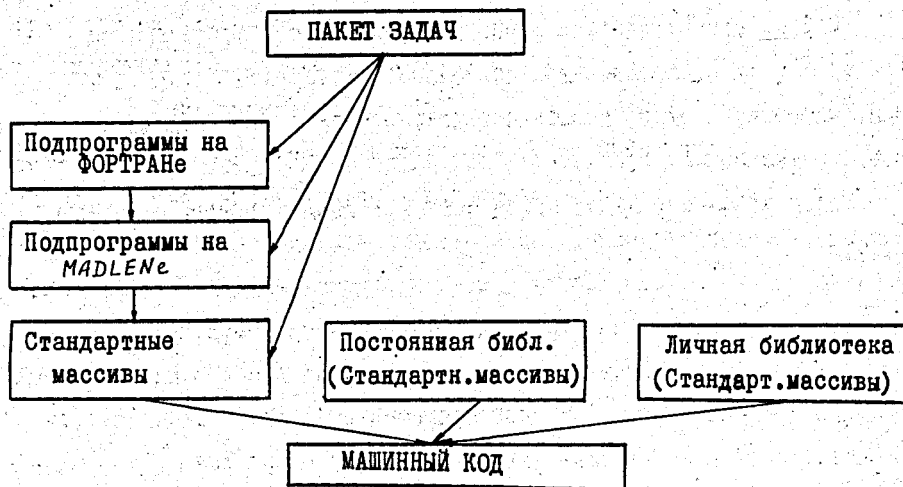
I. Общая организация перевода с машинного кода
в автокод

I.1. Постановка задачи

Благодаря усилиям, в основном, сотрудников ЛВТА, на ЭВМ БЭСМ-6 создана ныне действующая система автоматического программирования. Система включает трансляторы с ФОРТРАНа и автокода *MADLEN*, загрузчик, монитор, развитую библиотеку подпрограмм. Основным преимуществом системы является наличие двух уровней библиотеки на магнитных носителях информации, а именно: постоянной библиотеки на МБ и личной на МЛ. Причем оба уровня содержат подпрограммы, уже транслированные с автокода или ФОРТРАНа в стандартные массивы (называемые часто также модулями загрузки, или подпрограммами на языке загрузки, или подпрограммами в виде *BINARY*).

Весь процесс работы в режиме системы грубо можно представить схемой:

Схема I.



Пакет задач пользователя может содержать смесь подпрограмм, написанных на ФОРТРАНе и MADLENe, или полученных ранее в виде стандартных массивов. Настоящий вариант системы не представляет достаточно простой возможности ни для включения подпрограмм, написанных в машинном коде, в пакет задач, ни для записи их в библиотеку.

Ориентировка на автоматическое программирование является безусловно правильной и заслуживает всяческой поддержки. Тем не менее, в условиях БЭСМ-6 мы сталкиваемся с тем фактом, что (в основном, в других организациях) уже создана масса программ в машинном коде. Кроме того, имеются системы автоматического программирования (трансляторы и др.) с прямым выходом на машинный код, на программы в "абсолютных адресах".

Вместе с тем развитие математического обеспечения ЭВМ на современном этапе невозможно без активного обмена алгоритмами и программами между организациями-пользователями. Хотелось бы предоставить программистам, работающим в режиме мониторной системы "Дубна", возможность для использования (в частности, для включения в библиотеку) с минимумом непроизводительных затрат человеческого труда (типа переписывания алгоритма на автокод или ФОРТРАН) уже реализованных в машинном коде программ.

Основную часть этого труда, как будет показано в дальнейшем, может взять на себя вычислительная машина. Итак, стоит задача: получить из программы, написанной в машинном коде, программу в виде стандартного массива, т.е. программу в условных адресах, которая будет понята загрузчиком мониторной системы.

Наивно полагать, что это можно сделать без знания дополнительной информации о данной программе. Вместе с тем, оценить, была ли достаточной заданная информация о программе, программист может, лишь обзрев то, что у него получилось после перевода,

т.е. практически на уровне отладки переведенного. Отлаживать и обозревать стандартный массив нельзя, поэтому приходится делать следующий шаг, т.е. ставить задачу о переводе программы из машинного кода в автокод.

Очевидно, это задача, обратная трансляции программы с автокода и загрузке ее в память. В дальнейшем мы будем называть для краткости процесс перевода с машинного кода в автокод детрансляцией.

1.2. Общая организация детрансляции

Пользователь, желающий организовать детрансляцию подпрограмм или их частей из машинного кода в автокод *MADLEN*, получает в свое распоряжение подпрограмму *TØMADL*, описанную в этой работе.

Подпрограмма *TØMADL* написана на автокоде *MADLEN*. Она имеет собственную длину (без общих массивов) около 1300В слов. Подпрограмма *TØMADL* не имеет параметров. Вся входная информация к ней задается, как это показано в разделе II, посредством общих блоков задачи детрансляции.

Изготовив строку автокодной информации, *TØMADL* обращается к подпрограмме выдачи строки *STRPRN*, которая должна быть написана пользователем. В простейшем случае, если требуется лишь отпечатать заготовленную строку, подпрограмма *STRPRN* будет выглядеть так:

```
SUBROUTINE STRPRN (STRING)
  DIMENSION STRING(14)
  PRINT 1, STRING
1  FORMAT (IX, 13A6, A2)
  RETURN
END
```

Чаще бывает необходимо записать полученный автокодный текст на МЛ или отперфорировать его. Тогда рекомендуется написать еще две подпрограммы: подпрограмму заполнения и выдачи буфера и подпрограмму его записи на МЛ или перфорации $T\emptyset TAPE$.

Тогда общая организация детрансляции выразится схемами 2 + 6.

Схема 2

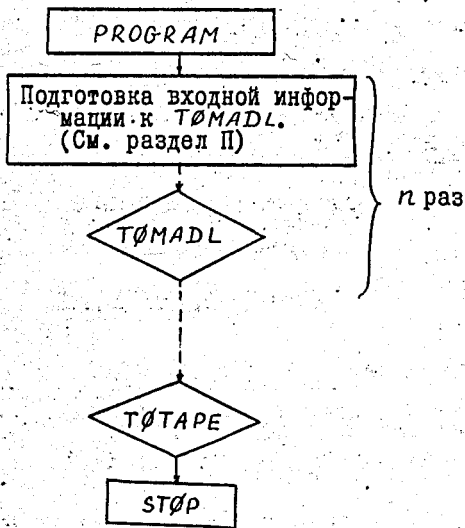


Схема 4

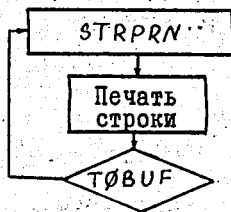


Схема 5

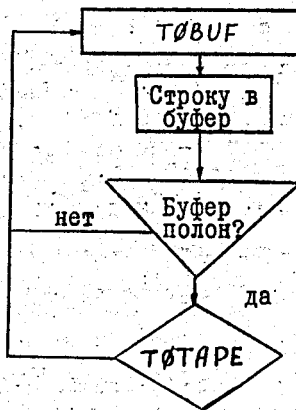


Схема 3

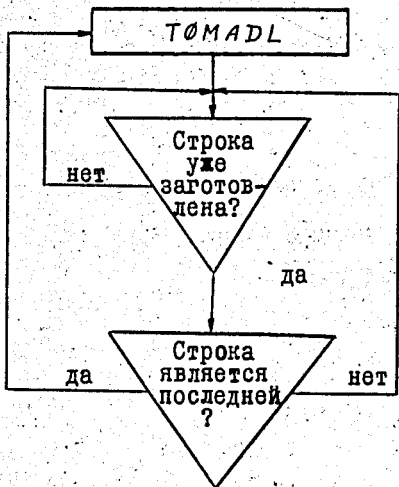
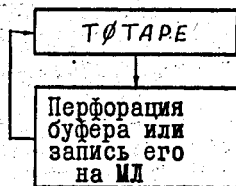


Схема 6



1.3. Редактирование

Полученные после детрансляции подпрограммы или их части могут содержать естественные ошибки, связанные с недостаточностью заданной для *TOMADL* информации или с трудностью полной автоматизации решения обратной задачи автокода и загрузчика.

В некоторых случаях бывает полезно, записав полученный при детрансляции текст на МЛ, добиваться его совершенствования с помощью аппарата редактирования (см. /2/).

В связи с этим следует отметить, что кроме обычных управляющих карт редактирования **DELETE* (исключить строки), **INSERT* (вставить после исключенных строк следующие за управляющей картой), **INSERT BELOW* (вставить строки после следующей строки).

В.Ю.Веретенным в настоящее время вводятся в употребление управляющие карты редактирования **PART* и **PART1*, которые позволяют редактировать строку по частям. Принцип использования этих карт следующий.

а) **PART*

Последовательность карт

	19 позиция	25 позиция
* <i>PART</i>	<i>N1</i>	<i>N2</i>
└ AA - BB		
└ ЧТО-НИБУДЬ		

означает: вставить со строки с номером *N1* по строку с номером *N2* включительно с позиции с десятичным номером AA по позицию с десятичным номером BB символы ЧТО-НИБУДЬ.

б) ж PART1

Последовательность карт

ж PART1	19 п. ↓ N1
┌ AA - BB	
┌ Символы, вставляемые в первую редактируемую карту	
┌ Символы, вставляемые во II-ю редактируемую карту	

┌ Символы, вставляемые в последнюю редактируемую карту
--

означает: начиная с карты с номером N1 вставить в позиции AA + BB указанную последовательность символов.

Под номером N1 понимается значение последних 8 символов карты.

П. Возможности подпрограммы TØMADL
и входная информация к ней

2.1. Минимальная информация для работы TØMADL. Команды.

Назовем объектом подпрограмму либо часть ее, написанную в машинном коде и занимающую ячейки с ННННтой по КККККую включительно, где ННННН и ККККК - восьмеричные адреса. В дальнейшем мы будем называть ячейки ННННН + ККККК полем объекта.

Предположим теперь, что объект должен быть каким-то образом обработан (например, детранслирован) в режиме мониторинга системы. Совершенно ясно, что для этого он должен быть введен в оперативную память машины (например, по экстракоду 073 с перфокарт, либо считан с МБ или МД) и размещен в некотором буфере, который мы будем называть буфером объекта. Буфер объекта не совпадает, вообще говоря, с полем объекта, но может с ним и совпадать.

Подпрограмма TØMADL считает буфером объекта общий блок /SUBR/SUBR (6I44). Программа, вызывающая TØMADL, должна позаботиться, чтобы к моменту вызова объект действительно был считан в буфер /SUBR/. Подпрограмме TØMADL должна быть сообщена информация о поле объекта засылкой в общую ячейку /FIELD /FIELD восьмеричного слова ННННН 000 000 ККККК, где ННННН и ККККК понимаются в указанном выше смысле.

Кроме того, вызывающая программа должна позаботиться об обозначении встречающихся в объекте индексных регистров, что целиком лежит на ее совести, даже в том случае, если индексные регистры должны быть обозначены своими десятичными номерами. Для этого используется общий блок /IR /IR (33), куда в кодировке ISØ должны быть посланы обозначения индекс-регистров, причем обозначение индекс-регистра i должно быть послано в ячейку с относительным адресом i этого блока.

Пример.

COMMON /IR/IR (32)

DATA (IR = '1', '1', '2', '1', '4', '4', '6', 'BASE', '8', ...)

В результате детрансляции в этом случае команды с индекс-регистрами $i = 1, 2, 4, 6, 8$ будут иметь в качестве модификатора индекс-регистр с тем же номером; команды с $i = 0, 3$ не будут иметь модификатора; команды с $i = 7$ будут иметь модификатор *BASE*; в командах с $i = 5$ номер индекс-регистра изменится и будет равным 4.

Не следует забывать, что кроме индексных регистров I + I7B, доступных математику, ЭВМ БЭСМ-6 имеет также индекс-регистры 20B + 37B, которые могут использоваться в диспетчерском режиме. Последние рекомендуется задавать их восьмеричными номерами; так, при $i = 34B = 28$, например, в соответствующее информационное слово IR (29) должно быть занесено значение '34B'.

Указанная выше информация является минимальной и обеспечивает работу *TOMADL* в следующем режиме:

а) Объект считается частью подпрограммы, то есть при детрансляции заголовков и описания не выдается.

В конце детранслированного объекта не выдается оператор *END*.

б) Все слова объекта детранслируются в пары команд. Исключения составляют некоторые случаи, когда слово по своей структуре не может быть парой команд (см. 3.4.1).

в) Если значение адресной части команды лежит в поле объекта, она заменяется при детрансляции автоматически сгенерированным идентификатором (см. 2.2).

В противном случае сохраняется восьмеричный адрес.

г) Если на слово есть ссылка (т.е. его адрес встретился в качестве адресной части какой-либо команды), оно при детрансляции метится соответствующим идентификатором.

2.2. Автоматическая генерация и наследственное задание идентификаторов

Обычной является автоматическая генерация идентификатора из адреса, которая производится следующим образом:

15 разрядов адреса разбиваются справа налево на четыре части: A_1 (3 разряда), A_2 (4 разряда), A_3 (4 разряда), A_4 (4 разряда). В соответствии с четырьмя выделенными кодами A_i , как это показано в следующих таблицах, генерируются буквы L_2, L_3, L_4 и цифры L_1 , которые и образуют идентификатор вида:

$$L_4 L_3 L_2 / L_1$$

A_1	0	1	2	3	4	5	6	7
L_1	I	2	3	4	5	6	7	8

A_2	0	I	2	3	4	5	6	7	10	11	12	13	14	15	16	17
L_2	K	L	M	N	Ø	P	Q	R	S	T	U	V	W	X	Y	Z
A_3	0	I	2	3	4	5	6	7	10	11	12	13	14	15	16	17
L_3	A	B	C	D	E	F	G	H	I	γ	K	L	M	N	Ø	P
A_4	0	I	2	3	4	5	6	7	10	11	12	13	14	15	16	17
L_4	G	H	I	γ	K	L	M	N	Ø	P	Q	R	S	T	U	V

Но иногда бывает необходимо, чтобы адрес детранслировался в некоторый заранее заданный идентификатор. Тогда в вызывающей TØMADL подпрограмме следует сформировать в общем блоке /IDENT/IDENT (3072) пару слов, первое из которых содержит соответствующий восьмеричный адрес в младших разрядах, а второе — идентификатор в коде ISØ. Разумеется, задаваемый идентификатор не должен состоять более чем из шести символов.

Следует учесть, что при этом адрес не обязательно должен принадлежать полю объекта, что часто бывает удобно. Кстати, в последнем случае с помощью общего блока / IDENT / можно задавать не только идентификаторы, но и многие другие адресные конструкции, так как тогда они не попадут ни в одну из меток подпрограммы. Суммарная длина такой конструкции также не должна превышать 6 символов.

Пример.

Если индексные регистры заданы своими десятичными номерами (см. 2.1) и вызывающая подпрограмма содержит операторы:

COMMON IDENT IDENT (3 072)

DATA (IDENT = 10001B, 'LABEL', 7777B, ' ',
40006B, 'BETA-5', 7000B, '= I - 1', 20B, '999'),

то следующий объект

I0 000 * I0 25 00020

I6 24 77000

I0 001 * 00 22 40006

I6 010 0001

I0 002 * 00 22 07000

00 26 10000

I0 003 * 00 26 10000

I6 37 10001

детранслируется в следующий участок подпрограммы:

ØAK/1 : 8, UTM, 999

I4, VTM, 77000B

LABEL : , UTC, BETA - 5

I4, XTA, 1

, UTC, = I - 1

, UZA, ØAK/1

I4, VLM, LABEL

2.3. Константы и массивы рабочих ячеек типа BSSN

Указанная в 2.I информация обеспечивает возможность нормальной работы детранслированного объекта после того, как он будет снабжен заголовком и концом (что можно сделать средствами 2.9), транслирован через автокод *MADLEN* и загружен в память на свое обычное место, т.е. в поле объекта.

Поставим задачу с наибольшей вероятностью добиться перемещаемости объекта, то есть его нормальной работы, если после всех указанных преобразований он будет загружен не в поле объекта, а в какое-нибудь другое место памяти. Основной причиной возможной неперемещаемости объекта является 2.I. б). Поэтому рекомендуется позаботиться о задании областей констант во входной информации к *TOMADL* .

Для этой цели используется общий блок /*ФСТ* /*ФСТ* (2048). Каждая область констант задается одним восьмеричным словом вида *AAAAA 000 000 BBBB*, где *AAAAA* - восьмеричный адрес первой константы, а *BBBB* - восьмеричный адрес последней константы этой области.

Разумеется, область должна быть непрерывной, т.е. не содержать внутри себя других объектов, например, команд. Иногда делается исключение для рабочих ячеек, которые можно задавать в виде констант.

Пример

Пусть в объекте с полем объекта *IO 000В + I7 777В* константы расположены в ячейках: *IO I20В + IO I37В*, *IO 222В*, *IO 373В*, *I7 000В + I7 777В*. Тогда в вызывающей подпрограмме задачи детрансляции, если она написана на ФОРТРАНе, должны присутствовать операторы:

СФММФН/FIELD/FIELD/ФСТ/ФСТ (2048)
DATA (FIELD = 10 000 000000 17 777В)

(ФСТ= 10 120 000000 10 137В, 10 222 000
000 10 222В, 10 373 000000 10 373В, 17 777 000
000 17 777В)

Каждая константа перейдет в автокодную конструкцию

< имя > : , ФСТ , < константа > .

Массивы рабочих ячеек задаются аналогично, но вместо / ФСТ /
в этом случае используется общий блок /BSSN /BSSN (256).

Разница будет в том, что тогда каждая область перейдет в
массив рабочих ячеек, задаваемый в автокоде оператором:

< метка > : , BSS , < длина массива > .

2.4. Специальные массивы

Иногда бывает, что в объекте ведется работа с ячейкой или
с массивом, лежащим в специальной области памяти (адреса
00 000 + 07 777В), причем их расположение в этой области существенно
(используются, например, в командах с коротким адресом). Тогда
имеет смысл детранслировать их в автокодные массивы типа LS. Для
этого существует общий блок /LS /LS (128). Информация должна
быть заслана в него в виде ААААА 000000 БББББ, где ААААА - восьме-
ричный адрес начала, а БББББ - восьмеричный адрес конца соответ-
ствующего специального массива.

Необходимо учесть, что специальные массивы извлекаются из
занимаемого ими прежде места в объекте и объект "сжимается" на
величину вынутых массивов, что может иногда вызвать в объекте
диспропорции, связанные с употреблением в командах относительных
адресов.

Если объект является подпрограммой (см. 2.9), то выдаются описания всех специальных массивов; в противном случае информация о специальных массивах используется только при обработке команд с коротким адресом, а также для изъятия этих массивов из тела объекта.

Бывает, что один и тот же адрес почти равновероятно выступает как в роли относительного, так и в роли абсолютного и трудно предположить - детранслировать ли его в идентификатор или оставить восьмеричным адресом. Обычно в таких случаях записывают детранслированный объект на МЛ и устраняют неправильности или улучшают его редактированием (см. 1.3).

Если адрес принадлежит некоторому специальному массиву и не принадлежит полю объекта, без редактирования можно обойтись, частично поступившись перемещаемостью подпрограммы. А именно: можно задать идентификатор в /IDENT/ (см. 2.2) для замены всюду этого адреса и "заэквивалентить" заданный идентификатор в /EQU/ (см. 2.6) своему адресу. Разумеется, тогда описывать его в /LS/ не надо.

Пример. Если в вызывающей подпрограмме имеются операторы:

```
COMMON /NAME/NAME/EQU/EQU/128/  
COMMON /IDENT/IDENT(3072)/FIELD/FIELD  
DATA (FIELD = 00 777 , 0000000 I777B)  
DATA (EQU='A', '40B', 2000B)  
DATA (IDENT = 00777B, 'PROGR', '40B', 'A', 2000B, 'B')  
DATA (NAME = 00777 000 000 00777B),
```

то в детранслируемом объекте выдадутся операторы:

```
PROGR : , NAME ,  
A : , EQU , 40B  
B : , EQU , 2000B.
```

и в адресных частях команд адрес 00 040B заменится идентификатором А, а адрес 02 000B заменится идентификатором В. Необходимость употребления здесь общей ячейки / NAME / будет пояснена в 2.9.

2.5. Общие массивы и общие блоки

Может оказаться, что объект должен детранслироваться в подпрограмму, имеющую общие ячейки, общие массивы или общие блоки с некоторыми другими подпрограммами.

При детрансляции соответствующие ячейки, массивы, блоки будут описаны как автокодные массивы LC конструкциями

`< имя > : , LC , < длина массива > ,`

если при вызове `TOMADL` в общем блоке `/LC/LC (I28)` будет задана информация о них аналогично 2.4.

При этом идентификаторы соответствующих массивов должны быть обязательно заданы для `TOMADL` с помощью общего блока `/IDENT` (см. 2.2).

При задании информации об общих блоках следует учесть, что фортранному блоку `/ИМЯ/` соответствует автокодный массив с идентификатором `ж ИМЯ ж`.

Отдельные массивы общего блока можно задать с помощью автокодных операторов `EQU` (пункт 2.6).

Пример

Фортранный оператор

`COMMON /A/ B, A(3), C(90), D (1000)`

полностью эквивалентен набору автокодных операторов:

`ж A ж : , LC , 2106B`
`B : , EQU , ж A ж`
`A : , EQU , ж A ж + I`
`C : , EQU , ж A ж + 4`
`D : , EQU , C + 90,`

которые могут быть получены средствами 2.5 и 2.6.

Общие массивы также извлекаются из занимаемого ими прежде места в объекте и объект "сжимается" на величину вынутых массивов. Последнее относится и к объектам, являющимся частями подпрограмм, но описания общих массивов для таких объектов не выдаются.

2.6. Эквивалентности

Эквивалентности задаются парами слов в кодировке $IS\emptyset$ в общем блоке $/EQU/EQU(I28)$.

В случае, если объект является подпрограммой, в результате детрансляции на каждую такую пару выдается автокодный оператор

$\langle \text{слово 1} \rangle : , EQU , \langle \text{слово 2} \rangle .$

Пример

Операторы:

```
COMMON /EQU/EQU(I28)
DATA (EQU='A', 'B-5', 'C', '4002B', 'D', '9', 'ALP', '=I-1')
```

иницируют в детранслированной подпрограмме:

A : , EQU , B - 5 .

C : , EQU , 4002B

D : , EQU , 9

ALP : , EQU , = I - 1 .

Если объект не является подпрограммой, содержимое блока $/EQU/$ на ход детрансляции не влияет.

2.7. Вызовы внешних подпрограмм и функций с помощью оператора CALL

Для того чтобы вызвать внешнюю подпрограмму или функцию (передать управление на ее начало с возвратом по содержимому I3 индексного регистра), в автокоде MADLEN существует, в частности, оператор *CALL*. Его преимуществом является то, что он служит одновременно описанием вызываемой подпрограммы или функции.

Возможность такого вызова из детранслированного объекта также предусмотрена. Чтобы обеспечить этот вызов подпрограммы или функции с передачей управления на вход, который имел в объекте адрес AAAAA, необходимо в общий блок */CALL/CALL(64)* задачи детрансляции занести слово AAAAA 000 000 AAAAA B (например, с помощью оператора *DATA*).

Пример

Пусть индексные регистры заданы в */IR/* своими десятичными номерами и поле объекта задано. Задача детрансляции с операторами

```
СФММФН /IDENT / IDENT (3072)
```

```
СФММФН /CALL / CALL (64)
```

```
DATA (IDENT= I0001B, 'FSIN', I0002B, 'FCFS')
```

переведет следующие команды:

```
40 000 * 00 30 I0001
```

```
15 3I I0001
```

```
40 001 * 00 30 I0002
```

```
15 3I I0002
```

в автокодные операторы:

```
,UJ, FSIN
```

```
,CALL, FSIN
```

```
,UJ, FCFS
```

```
13, VJM, FCFS
```

Идентификатор соответствующего входа в подпрограмму или функцию должен быть обязательно занесен в */IDENT /*.

2.8. Входы (ENTRY) в детранслируемую подпрограмму

Имеется возможность (аналогичная описанной в пункте 2.7) задать информацию о входах в детранслируемую подпрограмму. Для этого существует общий блок / ENTRY / ENTRY (64). Тогда в соответствующих ее местах TØMADL вставит автокодные операторы

< имя > : , ENTRY , .

При этом имя входа должно обязательно иметься в / IDENT /.

2.9. Заголовок детранслируемой подпрограммы или функции

Является ли детранслируемый объект подпрограммой или нет, целиком определяется состоянием общей ячейки / NAME / NAME .

В случае, если детранслируемый объект не является подпрограммой, а является лишь частью таковой, / NAME / должна содержать 0 во всех разрядах (т.е. действительный 0.). В этом случае выполняется 2.1. а).

Если же объект - подпрограмма (или функция), / NAME / должна содержать восьмеричную величину AAAAA OOOOO AAAAA, где AAAAA - адрес начала подпрограммы или функции.

Следует иметь в виду, что адрес начала подпрограммы может не совпадать ни с одним из адресов передачи управления в нее; и в этом случае он должен быть занесен в / NAME /, а адреса передачи управления - в / ENTRY /. Адрес начала подпрограммы, заносимый в / NAME /, всегда должен совпадать с адресом, содержащимся в 15 старших разрядах ячейки / FIELD /.

При / NAME / ≠ 0 выдается заголовок подпрограммы в виде:

< имя > : , NAME , .

Затем, если они имеются, описания (LC , LS и EQU).

В конце подпрограммы выдается оператор , END , .

Как и имя каждого входа, имя подпрограммы или функции должно быть задано в IDENT .

2.10. Комментарии к детранслируемому объекту

TOMADL записывает комментарии в 43 + 72 позиции строки автокодного текста. В обычном режиме туда пишется адресная часть детранслируемой команды в восьмеричном виде, но программист может вносить туда и более разумные комментарии.

Делать это удобно с помощью подпрограммы *COMENT*, которую программист должен написать на автокоде и которая должна состоять из одних констант. Вид этой подпрограммы следующий:

```
      *COMENT* : NAME ,  
информ.      , LOG , AAAAA  
единица      , ISØ , 60H      комментарий I      комментарий 2  
                                     30 символов      30 символов  
      , LOG , 00000  
      , END ,
```

Здесь информационная единица содержит информацию о комментариях к одному машинному слову.

AAAAA - восьмеричный адрес этого слова.

Если слово детранслируется в пару команд, левая (верхняя) команда комментируется с помощью "комментария I", а правая (нижняя) - с помощью "комментария 2".

Если слово детранслируется в константу, "комментарий 2" проигнорируется и в строку будет внесен "комментарий I".

Информационные единицы о комментариях должны быть расположены в подпрограмме *COMENT* в порядке возрастания восьмеричных адресов AAAAA.

ЛИТЕРАТУРА

1. А.И.Волков. "Автокод MADLEN ". Б4-II-4654, Дубна, 1969.
2. В.Д.Веретенков, Н.С.Зайкин, В.А.Ростовцев, А.Фомичева.
Мониторная система БЭСМ-6. Редактирование текстовой информации.
Материалы совещания по программированию и вычислительным методам решения физических задач. Программирование и вычислительные методы. II-4655, Дубна, 1969.
3. В.П.Шириков. Математическое обеспечение БЭСМ-6. Программирование и вычислительные методы. II-4655, Дубна, 1969.
4. И.Н.Силин и др. Мониторная система БЭСМ-6. Общая организация. Программирование и вычислительные методы. II-4655, Дубна, 1969.

Рукопись поступила в издательский отдел

27 февраля 1970 года.