

48408

Г-15

30/III-71

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

3044/2-71



10 - 5911

ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
И АВТОМАТИЗАЦИИ

В.В. Галактионов

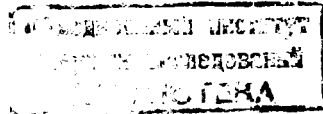
ТРАНСЛЯТОР НА БЭСМ-6
С АВТОКОДА SLANG 3 ДЛЯ ТПА

1971

10 - 5911

В.В. Галактионов

ТРАНСЛЯТОР НА БЭСМ-6
С АВТОКОДА SLANG 3 ДЛЯ ТПА



С точки зрения логической структуры ЭВМ ТПА, область применения ее может быть самой разнообразной. ТПА можно использовать в качестве управляющего элемента в системе автоматического управления, а также как универсальную вычислительную машину общего назначения. Простота и гибкость структуры, сводящие к минимуму затраты, связанные с организацией и изменением системы, а также наличие программного канала, способного обслуживать до 64 совместно работающих устройств, позволяют наиболее эффективно использовать ТПА именно в качестве управляющего элемента некоторой системы.

Эффективное использование любой ЭВМ возможно только при наличии развитого математического обеспечения этой ЭВМ.

Существующее матобеспечение ТПА состоит из неподвижного ассемблера *SLAN61*, перемещаемого *SLAN62* и различных загрузчиков (*RIM-loader*, *BIN-loader*) (см. /1/).

Перфокарочный ввод, а, следовательно, и текстовые и двоичные перфокарты существенно затрудняют программирование на языках указанных выше ассемблеров.

Необозримость программы на перфокарте, необходимость перебивки всей перфокарты при малейшем изменении или исправлении программы, а также трехпроходная трансляция (надо трижды вводить текстовую перфокарту) должны предполагать наличие большого количества свободного времени и крепких нервов у программиста, работающего с достаточно большими программами.

Отсутствие того и другого у автора этой работы и явилось причиной написания транслятора *SLAN63* на БЭСМ-6 для ТПА. Вся входная текстовая информация пробивается на перфокартах на *1CT* или УПП-2. Результат трансляции на БЭСМ-6 - листинг с текстом программы

и восьмидорожечная перфолента с двоичной программой в *BIN*-формате. Полученную перфоленту можно вводить на ТПА двоичным загрузчиком (*BIN-loader*). В основу языка *SLANG3* положен язык *SLANG1* с некоторыми добавлениями и расширениями. Введены автоматическая косвенная адресация, группа *BSS*, текстовые константы.

Транслятор, в основном, написан на автокоде МАДЛЕН^{2/3}, управляющая часть его - на ФОРТРАНе. Подпрограммы транслятора собраны в личную библиотеку на магнитной ленте БЭСМ-6.

Автор благодарит В.П.Ширкова за помощь, оказанную им при редактировании этой работы.

Основные символы

Язык автокода **SLANG3** строится из следующих основных символов:

Буквы: 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.

Цифры: 1, 2, 3, 4, 5, 6, 7, 8, 9, 0.

Разделительные знаки:

<u>Символы</u>	<u>Употребление</u>
␣ пробел	разделитель
+ плюс	знак константы либо знак сложения
- минус	знак константы либо знак вычитания
, запятая	признак метки
= равенство	параметрическое присваивание
* звездочка	указатель адресного кода
\$ доллар	указатель конца программы
. точка	указатель адреса текущей ячейки
/ SLASH	признак комментария
' апостроф	разделитель операторов

В комментариях допускается использование всех букв русского алфавита.

Элементарные конструкции языка включают в себя числа, идентификаторы, константы.

Числа:

целое без знака - последовательность четырех любых восьмеричных цифр.

Примеры: 1234
0
7777

Целое имеет вид:

$\pm n$,

где n - целое без знака. Знак + может быть опущен.

Примеры: II

-II3

+77

Отрицательные числа представляются в дополнительном коде.

Идентификаторы.

Идентификатор - последовательность букв и цифр, начинающаяся с буквы.

Идентификатор не может содержать более чем 6 символов.

Идентификатор должен оканчиваться неалфавитным символом (пробелом, запятой, знаками "+" или "-" и др.).

Примеры: A, JMS

EXIT

SLANG1

Константы.

Ассемблер допускает три типа констант:

восьмеричные,

текстовые,

символические.

Восьмеричные и символические константы транслируются в одно машинное слово.

Восьмеричная константа - последовательность не более чем четырех восьмеричных цифр. Недостающие до полного машинного слова цифры будут дополнены нулями слева.

Пример.

Константы

Слова ТПА

7777

7777

I

000I

0

0000

К восьмеричным константам относятся константы типа:

. \pm восьмеричная константа, (.) - указывает адрес текущей ячейки.

Примеры: .+IO
.-I

Текстовые константы записываются в следующем виде:

TEXT *n* HXXX...X
n символов

n - восьмеричное целое без знака число.

При трансляции такой текстовой константы отводится *n* слов для нее по одному символу в слове. Символы записываются в коде ASCII, см. приложение I.

Пример:

TEXT SHWELL.

После трансляции получим:

	0327
	0305
	0314
	0314
	0256

Символические константы имеют следующую структуру:

$\pm A \pm B$,

где A и B - идентификаторы либо числа.

Пример: A + B
C + IO
D - IO
ALPHA
BETA - 7000

Команды.

Три младших разряда машинного слова определяют код операции



В случае адресной команды (коды 0-5), третий разряд является признаком косвенной адресации. I в 3 разряде означает, что полный адрес команды находится в ячейке, адрес которой указан в 5-II разрядах данной команды. I в 4 разряде означает, что 5-II разряды определяют относительный адрес в текущей странице. Если в 4 разряде ноль, семиразрядный адрес выбирается из нулевой страницы. Наличие косвенной адресации и указателя страницы объясняется малой разрядностью слова ТПА, вследствие чего 7 разрядов адреса не могут указать адреса произвольной ячейки памяти. Отсюда и листовая (по 200₈ слов) структура памяти.

Мнемонические коды команд приведены в приложении П.

Шестерка в 0-2 разрядах слова указывает на команды работы с внешними устройствами.

Семерка в этих разрядах - признак безадресной команды.

Метки:

Идентификатор, ограниченный запятой, определяет адрес текущего слова и называется меткой.

Примеры:

A, +100
RETURN, JMS I RET
CLEAR, CLA

Полный адрес имеет структуру:

A ± B, где A и B метки или определенные каким-либо другим образом идентификаторы.
B может быть восьмеричным числом.

Примеры:

JMS RET + 11
TAD ALPHA - BETA

Простой адрес - адрес одной из следующих форм:

1. Идентификатор
2. Восьмеричное число
3. \pm восьмеричное число.

Точка в адресной части означает адрес текущей ячейки.

Примеры:

```
TAD IO
DCA I5
JMP .-2
```

Автоматическая индексация

Свойством автоматической индексации обладают только ячейки I0-I7 нулевой страницы. Если к какой-либо из этих ячеек обратиться в режиме косвенной адресации, к ней автоматически добавляется единица, а лишь затем исполняется команда. Указанные ячейки имеют символические названия: *AUT I0*, *AUT I1*, ..., *AUT I7* соответственно.

Пример:

```
*I0
0077
:
*I000
CLA
TAD I 10
```

На сумматор ТПА вызовется содержимое ячейки I00.

I - признак косвенной адресации.

Параметрическое присваивание

Суть параметрического присваивания заключается в присвоении какому-либо идентификатору численного значения. Это очень удобно для введения новых команд, адресов или констант, поскольку этот оператор не попадает в оттранслированную программу.

Примеры:

```

ALPHA = 6I42
BETA = 6000
* BETA + 10
⋮
JMS ALPHA

```

Резервирование массивов

Зарезервировать массив длиной в N ячеек, не пробывая на перфокарте N констант, можно оператором $BSS\ N$.

Пример: $ALPHA, BSS\ 10$

В программе, начиная с ячейки $ALPHA$, будет размещено 10_8 нулей.

Автоматическая косвенная адресация (АКА)

При составлении больших программ (свыше 200_8 слов) всякие ссылки на идентификаторы, находящиеся в разных страницах, возможны только с использованием косвенной адресации. Для этого в текущей странице отводится ячейка, в которой программист заранее запасает адрес нужной ячейки из другой страницы. Автоматическая косвенная адресация устраняет это малоприятное обстоятельство. Для этого достаточно установить в определенный момент признак АКА с указанием массива для полных адресов. Это делается одним оператором типа:

$BASE\ * \ BUF$, где BUF — метка, указанного выше буфера для косвенной адресации.

Пример:

```

*400
⋮
BASE* BUFF
CLA
TAD A
DCA B
⋮
BUFF, BSS 10
⋮
* 7000
A, 707
B, 0
⋮

```

Таким образом, можно писать программу, не заботясь о том, что в одной из страниц встречаются ссылки на метки (*TAD A*, *DCA B*), расположенные в другой странице. Транслятор сам вставит в команду *TAD A* признак косвенной адресации ссылкой на одну из ячеек буфера *BUFF*. Единственное ограничение: буфер для ссылок должен находиться в той же странице памяти, что и команда с косвенной адресацией. Снимается признак АКА оператором

BAS ж .

Микропрограммирование

Некоторые безадресные операции могут выполняться одновременно и записываются они на языке *SLANG3* в одном операторе.

Пример: Оператор *CLA CLL* делает то же самое, что и два последовательных оператора

CLA
CLL

Специальные символы

1. Указатель текущего адреса.

Точка в адресной части или в константе определяет адрес ячейки этого слова ТПА.

Примеры:

JMP .+2
.+2400

2. Звездочка *ж* используется в качестве указателя адресного кода.

Адресный код *жA[±]C*, где *A* – ранее определенный идентификатор, *C* – восьмеричная константа,

определяет адрес загрузки следующей за ним части программы.

Примеры: #4000

A, CLA

:

#A+200

JMP A

3. Наклонная черта / (*slash*) является признаком комментария.

Пример:

CLA / THIS IS COMMENT.

4. Буква I в адресной части указывает на наличие косвенной адресации, поэтому использование ее в качестве метки нежелательно.

Пример:

CLA
TAD I AUTO1

:

AUTO1, 3000

5. Апостроф используется в качестве разделителя операторов SLANG 3, размещенных на одной перфокарте.

Пример:

#400 ' CLA ' TAD A ' DCA B

6. Знак # означает конец программы.
При пробивке текста программы на УПП-2, в качестве признака конца программы используется символ " ⧫ ".

7. Идентификатор, заканчивающийся запятой, является меткой.

8. Особую роль как разделителя играет пробел. Пробел разделяет коды операции, адресные части и метки. Количество пробелов несущественно, достаточно одного.

Диагностика при обнаружении ошибок

Транслятор с автокода обнаруживает некоторые формальные несоответствия синтаксису языка, ошибки в структуре операторов и печатает диагностические тексты. В зависимости от вида ошибки игнорируется либо вся команда или константа, либо некоторые их части. В случае обнаружения ошибки при трансляции, выдача двоичной программы на перфоленту не производится.

Диагностические тексты:

1. Неописанный идентификатор XXXXXX.
2. Дважды описанный идентификатор XXXXXX.
3. Несоответствие страниц.

Последняя диагностика выдается в случае появления в адресе команды метки не из текущей страницы.

Подготовка программы для трансляции

1. Текст программы пробивается на перфокартах на устройствах Ист или УПП-2.

73-80 позиции перфокарты при трансляции игнорируются.

Расположение текста оператора на перфокарте произвольное.

2. Заканчиваться программа должна знаком конца " / " или, в случае пробивки на УПП-2, символом " ◊ ".

3. Нужно сформировать следующий пакет перфокарт:

```
*NAME SLANG  
*ASSIGN LTAPE  
*PERSONAL LIBRARY  
*MAIN SLANG  
*EXECUTE
```

Пакет программы, написанной на SLANG3

```
*END FILE
```

Управляющие карты (со звездочками) пробиваются с первой позиции. В конце пакета должна находиться карта с диспетчерским концом БЭСМ-6 (сплошные пробивки в I и 4I позициях). В инструкции операторам БЭСМ-6 должны быть указания о постановке магнитной ленты с транслятором *SLANG3* и о включении ленточного перфатора.

Результат трансляции - листинг с двоичной программой и текстом ее, а также восьмидорожечная перфолента в *VIN*-формате. Полученную перфоленту можно вводить на ТИА двоичным загрузчиком (*VIN-loader*).

VIN-формат:

```

перфолента
10000. 000
10000. 000
01 X . Y      адресный код
00 V . Z
00 A . B
00 C . D      программа
00 A . B
00 C . D
:
00 A . B      контрольная сумма
00 C . D
10000. 000
10000. 000
  
```

12-разрядный адресный код	<u>000</u>	<u>000</u>	<u>000</u>	<u>000</u>
	X	Y	V	Z
12-разрядное слово ТИА	<u>000</u>	<u>000</u>	<u>000</u>	<u>000</u>
	A	B	C	D

В конце программы на листинге печатается таблица идентификаторов и их абсолютные значения.

Пример программы, написанной на SLANG3

Программа печатает на телетайпе:

THIS IS EXAMPLE .,

затем происходит возврат каретки телетайпа и пропуск строки.

```
BASE* BUF
#400 / ЭТА ПРОГРАММА - ПРИМЕР
CLA / РАБОТЫ С ТЕЛЕТАЙПОМ
TAD TEXT1 -2
DCA ALPHA -1
TAD TEXT1 -1
DCA ALPHA -2
JMS ALPHA
HLT
21
TEXT1 / СИМВОЛИЧЕСКАЯ КОНСТАНТА
TEXT1, TEXT 20THIS IS EXAMPLE.
2I2
2I5
#7000
DATA, 0 / АДРЕС МАССИВА ДЛЯ ПЕЧАТИ
0 / КОЛИЧЕСТВО СИМВОЛОВ
ALPHA, 0
CLA
TAD DATA +1
CMA
DCA DATA +1
TAD RET
TLS
CHECK, TSF
JMP. -1
CLA
TAD LNFD
TLS / ПЕЧАТЬ
PR, TSF
JMP, -1
CLA
TAD I DATA
```

```

    T L S
    I S Z   D A T A
    I S Z   D A T A + 1
    J M P   P R
    J M P   I   A L P H A
R E T , 2 1 5           / К О Д   В О З В Р А Т А   К А Р Е Т К И
L N F D , 2 1 2       / К О Д   П Р О П У С К А   С Т Р О К И
    $

```

ПРИЛОЖЕНИЕ I

Символы ASCII

Символ	8-разр. код	Символ	8-разр. код	Символ	8-разрядный код
A	30I	R	322	8	270
B	302	S	323	9	271
C	303	T	324	(250
D	304	U	325)	251
E	305	V	326	ж	252
F	306	W	327	+	253
G	307	X	330	,	254
H	3I0	Y	33I	-	255
I	3II	Z	332	.	256
J	3I2	0	260	/	257
K	3I3	I	26I	:	272
L	3I4	2	262	=	275
M	3I5	3	263	пробел	240
N	3I6	4	264	пропуск	2I5
O	3I7	5	265	строки	
P	320	6	666	возврат	2I5
Q	32I	7	267	каретки	

ПРИЛОЖЕНИЕ П

Таблица мнемочкодов команд

Адресные команды

Принятые обозначения: АС - сумматор, L - линк (дополнительный разряд сумматора).

Мнемочкод	Код	Содержание операции
AND Y	0	Логическое умножение содержимого сумматора на содержимое ячейки Y.
TAD Y	1	Сложение.
ISZ Y	2	Добавление 1 к ячейке Y и пропуск следующей команды по нулю в ячейке Y.
DCA Y	3	Запись слова в память.
JMS Y	4	Переход по адресу Y+1. В ячейке Y записывается адрес возврата.
JMP Y	5	Безусловный переход по адресу Y.

Безадресные команды

Мнемочкод	Код	Содержание операции
CLA	7200	Очистка сумматора АС.
CLL	7100	Очистка линка.
CMA	7040	Обращение сумматора. \overline{AC}
CML	7020	Обращение линка. \overline{L}
RAR	7010	Циклический сдвиг вправо сумматора и линка на один разряд.
RAL	7004	Циклический сдвиг влево сумматора и линка на один разряд.
RTR	7012	Циклический сдвиг вправо сумматора и линка на два разряда.

Мнемокод	Код	Содержание операции
<i>RTL</i>	7006	Циклический сдвиг влево сумматора и линка на два разряда.
<i>IAC</i>	700I	Добавление единицы к содержимому сумматора.
<i>CIA</i>	704I	Дополнение сумматора.
<i>STA</i>	7240	Засылка единиц во все разряды сумматора.
<i>STL</i>	7I20	Засылка единицы в линк.
<i>GLK</i>	7240	Засылка содержимого линка в одиннадцатый разряд сумматора.
<i>NOP</i>	7000	Пауза.
<i>SPA</i>	75I0	Пропуск команды, если $AC_0=0$.
<i>SMA</i>	7500	Пропуск команды, если $AC_0=I$.
<i>SNA</i>	7450	Пропуск команды, если $AC \neq 0$.
<i>SZA</i>	7440	Пропуск команды, если $AC=0$.
<i>SZL</i>	7430	Пропуск команды, если $L=0$.
<i>SNL</i>	7420	Пропуск команды, если $L=I$.
<i>SKP</i>	74I0	Пропуск команды.
<i>OSR</i>	7404	Логическое сложение сумматора с содержимым I2-разрядного пультового регистра ТПА.
<i>LAS</i>	7604	Засылка содержимого пультового регистра ТПА в сумматор.
<i>HLT</i>	7402	Останов.

Некоторые команды для работы с внешними устройствами

Мнемокод	Код	Содержание команды
<i>ION</i>	600I	Разрешение прерываний.
<i>IOF</i>	6002	Запрещение прерываний.

Работа с телетайпом ТТ

Мнемокод	Код	Содержание команды
KSF	6031	Пропуск команды, если флаг клавиатуры ТТ=1.
KCC	6032	Очистка сумматора и флага ТТ.
KRS	6034	Добавление к сумматору АС _{4-II} содержимого 8-разрядного регистра ТТ.
KRB	6036	Очистка сумматора и флага ТТ. Добавление к сумматору АС _{4-II} содержимого 8-разрядного регистра ТТ.
TSF	6041	Пропуск команды, если флаг перффоратора ТТ=1.
TCF	6042	Очистка флага перффоратора ТТ.
TPC	6044	Перфорация содержимого сумматора АС _{4-II} .
TLS	6046	Очистка флага перффоратора ТТ. Перфорация содержимого сумматора АС _{4-II} .

ЛИТЕРАТУРА

1. TPA. SLANG. PROGRAMMING LANGUAGE.

Munkaszám: KFKI 3819 Budapest, 1968.

2. А.И.Волков. Автокод МАДЛЕН. Б4-II-4654, Дубна, 1969 г.

Рукопись поступила в издательский отдел

2 июля 1971 года.