

сообщения <sup>e</sup>  
объединенного  
института  
ядерных  
исследований  
дубна

1661/2-81

30/н<sup>т</sup>-81

P11-81-51

Е.Ю.Мазепа

КРОСС-АССЕМБЛЕР ДЛЯ INTEL-8080  
НА ЭВМ ЕС-1010

1981

Как уже отмечалось в работе<sup>/1/</sup>, перспективы развития концентратора терминалов (ЕС-1010) предусматривают организацию связи ЕС-1010 с CDC-6500 (одной из базовых ЭВМ ЦВК ОИЯИ).

Основным блоком устройства связи ЕС-1010 с CDC-6500 был выбран микропроцессор INTEL-8085A, технически более совершенный и имеющий полную программную совместимость с известным микропроцессором INTEL-8080<sup>/4/</sup>. Технические условия организации связи ЕС-1010 и INTEL-8085 позволяют иметь микропроцессору собственную память в размере 4К байтов (K=1024). Так как в памяти INTEL-8085A предполагается разместить программы, поддерживающие протокол обмена ЕС-1010 с CDC-6500, то располагать в памяти микропроцессора транслятор с языка ASSEMBLER<sup>/4/</sup> не представляется возможным. С другой стороны, система математического обеспечения концентратора имеет развитые редактор текстов и файловую подсистему. Поэтому было решено транслировать программы для INTEL-8085A, написанные на языке ASSEMBLER, на ЭВМ ЕС-1010.

Программы, производящие такого рода трансляцию, обычно называют кросс-ассемблерами. Кросс-ассемблер, о котором идет речь в данной работе, написан на языке ASS2 и может быть использован на любой конфигурации ЭВМ ЕС-1010, оснащенной магнитным диском ЕС-5060<sup>/2/</sup> (мини-диск).

#### Логическая схема работы кросс-ассемблера

Кросс-ассемблер реализован по двухпроходной схеме трансляции<sup>/6/</sup>. Во время первого прохода строится таблица идентификаторов, в которую собираются определения идентификаторов, встреченные в транслируемой программе.

Второй проход выполняет, собственно, саму трансляцию, используя значения идентификаторов, полученные во время первого прохода.

Рассмотрим подробнее первый и второй проходы кросс-ассемблера.

Первый проход. Цель первого прохода мы уже сформулировали. Для достижения этой цели кросс-ассемблер должен просматривать поля меток и составлять по ним определения идентификаторов. Помимо этого на первом проходе должен осуществляться анализ программы для того, чтобы правильно вести счетчик размещения идентификаторов относительно начала программы, а также необходимо распознавать концы команд и понимать псевдоинструкции резервирования памяти и генерирования байтов данных.

На каждый вновь встреченный идентификатор, используя хеш-функцию <sup>3/</sup>, отводим 10 байтов следующим образом:

0	1	2	3	4	5	6	7	8	9
NAME					ADR		SP		

где

- NAME (6 байтов) - идентификатор в коде EBCDIC (если количество букв в идентификаторе менее шести, то правые байты заполняются пробелами);
- ADR (2 байта) - адрес, по которому встретился идентификатор;
- SP (2 байта) - информация о состоянии идентификатора (в описываемой версии использован только самый левый бит, единица в этом бите означает, что идентификатор дважды описан).

Приведем краткую блок-схему первого прохода кросс-ассемблера (рис.1).

Второй проход. Главной задачей второго прохода является составление двоичной версии программы вместе с информацией о ее настройке.

При обработке входной строки поле метки игнорируется, т.к. оно уже полностью обработано на первом проходе (однако инструкция печатается полностью). Код операции проверяется по таблице. Если такого кода нет, выдается сообщение об ошибке, в противном случае управление передается на подпрограмму, обрабатывающую данный код операции, где происходит обработка поля операнда. В процессе этой обработки, если необходимо, может быть и обращение к таблице идентификаторов.

Результатом работы второго прохода является сформированная в зоне `STGO` <sup>12/</sup> мини-диска двоичная версия программы вместе с информацией о ее настройке, а также листинг оттранслированной программы, если он был заказан.

Блок-схема второго прохода приведена на рис.2.

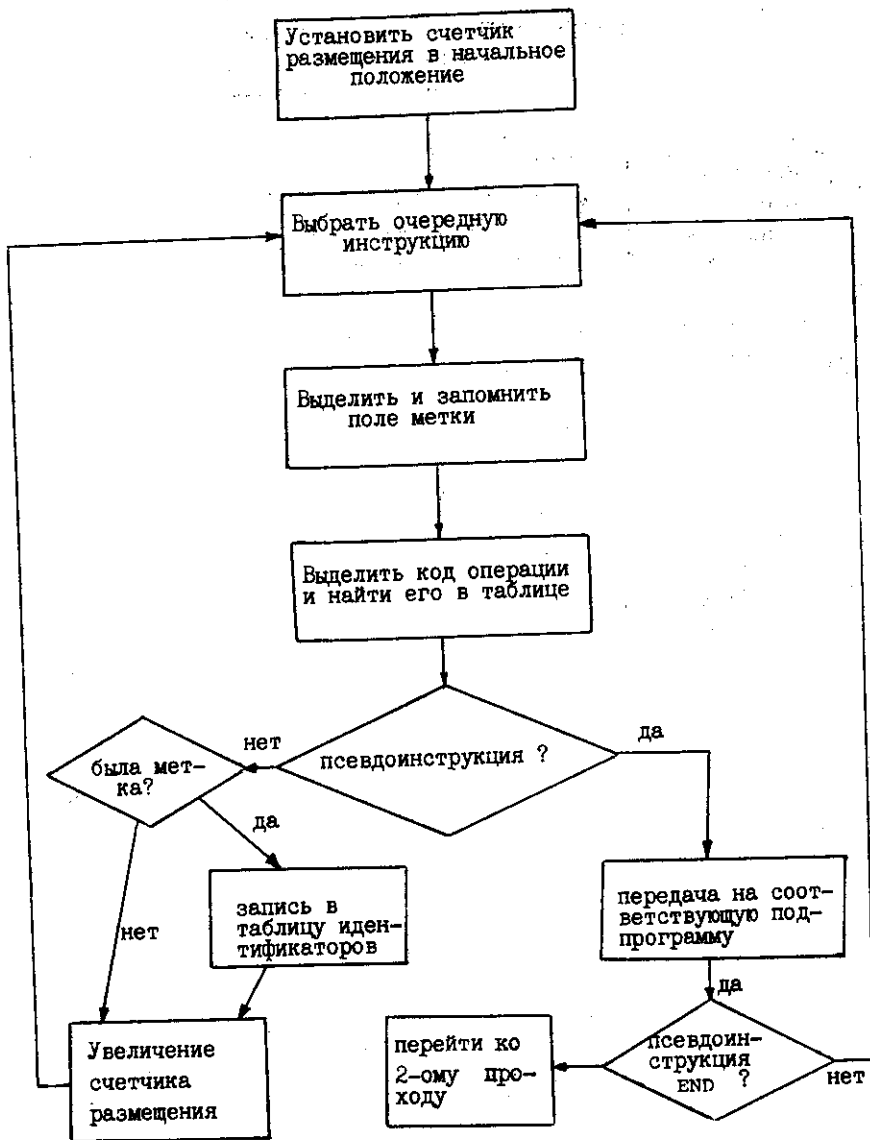


Рис.1. Блок-схема первого прохода.

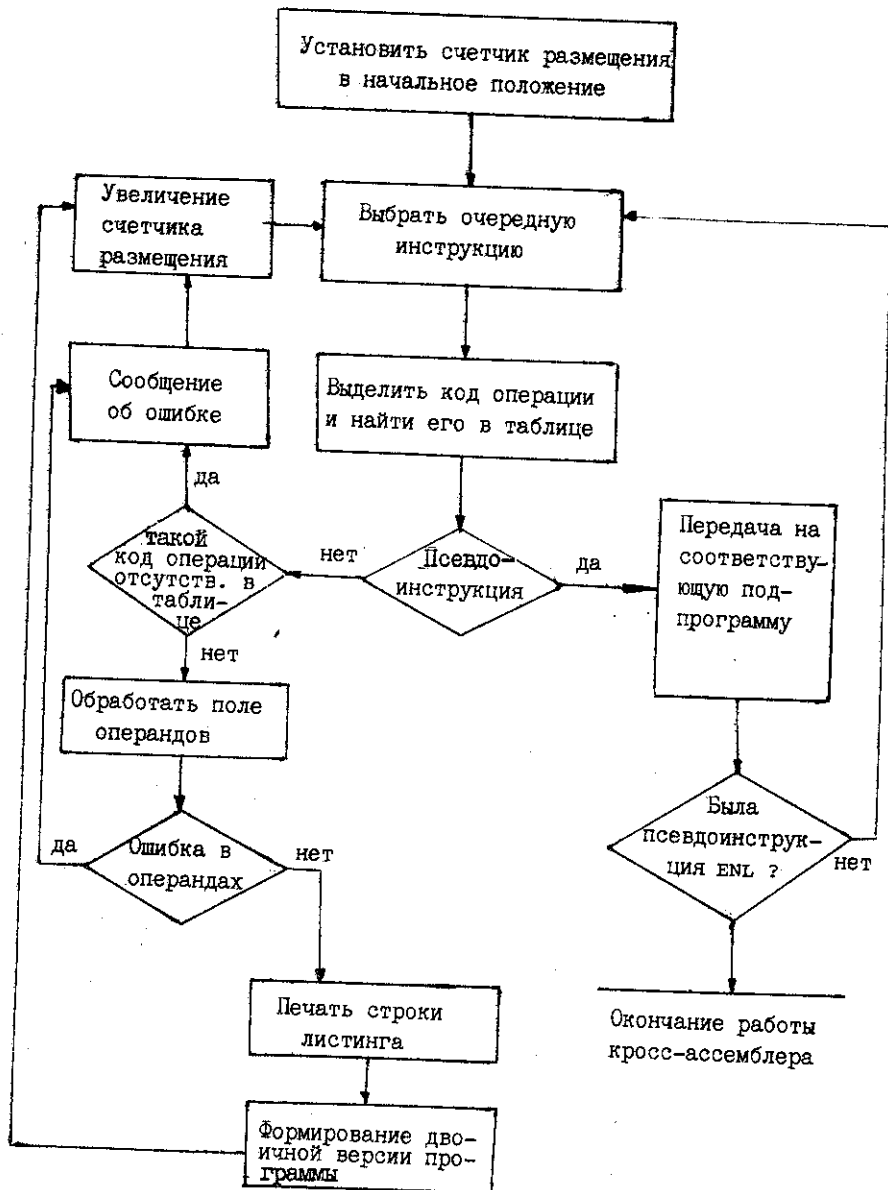


Рис. 2. Блок-схема второго прохода.

## Порядок работы с кросс-ассемблером

1. Пользователь, желающий работать с кросс-ассемблером, должен подготовить исходный текст своей программы, написанной на языке ASSEMBLER для INTEL-8080, на каком-нибудь внешнем носителе данных. Кросс-ассемблер может работать с исходным текстом, если текст подготовлен:

- а) на перфокартах (в коде КПК-12);
- б) в зоне ДА мини-диска по правилам стандартной для ЕС-1010 файловой системы RMS-M (в этом случае файл, содержащий исходный текст, должен быть по порядковому номеру первым файлом);
- в) как перманентный файл в системе концентратора терминалов.

### Замечание 1

Кросс-ассемблер работает по двухпроходной схеме, поэтому в связи с тем, что дисковая память на ЭВМ ЕС-1010 является "горячим" ресурсом (по крайней мере, это справедливо для системы концентратора), буферизация ввода на дисках не предусмотрена и в случае а колода перфокарт будет считываться два раза. Кроме того, в системе концентратора существует средство заведения перманентных файлов с перфокарт, поэтому случай а сводится к случаю в.

### Замечание 2

Существует программа PFTOBS, перемещающая с преобразованием форматов перманентный файл с диска ЕС-5053 в зону ДА мини-диска, поэтому случай в сводится к случаю б.

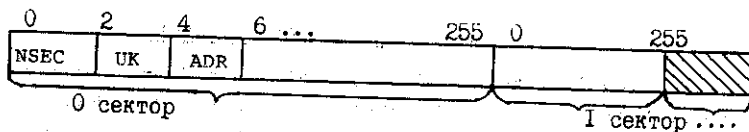
2. После того как исходный текст подготовлен на внешнем носителе данных, пользователь может приступить непосредственно к трансляции.

Кросс-ассемблер может, например, храниться в зоне ЕР мини-диска под именем INTEL. Тогда загрузка и запуск кросс-ассемблера осуществляется с устройства, закрепленного за операционной меткой M:OC приказом %CALL/INTEL/.

После этого программа INTEL запрашивает у пользователя устройство, с которого будет считываться исходный текст.

Выходными данными программы INTEL являются:

- 1) в зоне GIG0 модуль загрузки для INTEL-8080. Структура этого модуля следующая:



где

- NSEC (2 байта) – количество секторов, занятых модулем загрузки;  
 UK (2 байта) – количество байтов в последнем секторе;  
 ADR (2 байта) – адрес, с которого будет помещаться модуль загрузки в памяти INTEL-8080.

Остальные секторы модуля загрузки содержат двоичную информацию – результат трансляции;

- 2) листинг, который выводится на устройство, закрепленное за операционной меткой M:LO <sup>1/2</sup>. Каждая строка листинга может быть:

А. Диагностическим текстом, что означает ошибку в следующей инструкции.

Имеются следующие диагностические тексты:

1. Идентификатор длиннее 6-ти символов.
2. КОП длиннее 4-х символов.
3. Некорректный КОП.
4. Некорректный идентификатор.
5. Ошибка в операнде.
6. Дважды описан идентификатор.
7. Неописанный идентификатор.
8. Переполнена таблица идентификаторов, трансляция прекращена.

Б. Оттранслированной строкой инструкции (либо продолжением этой строки). Например:

```

1      FLAG      EQU      4000H
2                      ORG      1000H
3 1000 BA WAIT   LDA      FLAG
      1001 00
      1002 40
      1003 F1          POP      PSW
4                      END
  
```

В. Строкой, завершающей листинг:

число оп. – N ,

где N – число ошибок трансляции.

Ограничения и расширения языка ASSEMBLER для программы  
INTEL

1. Отсутствуют макросредства.
2. Отсутствует псевдоинструкция SET.
3. Отсутствуют средства условной трансляции.
4. Отсутствует адресная арифметика.
5. Допускается длина идентификаторов до шести символов.
6. Метка должна начинаться с первой позиции инструкции.
7. Тип констант может быть только десятичный или шестнадцатиричный.
8. Инструкция EQU применима только к адресам, и на месте операнда не может быть идентификатора.

В заключение автор хотел бы выразить искреннюю благодарность М.Ю.Попову и Е.Д.Федонькину за полезные обсуждения, а также В.Г.Сидорову за предоставленную возможность проверить работу программы INTEL на системе, созданной в ЛЯП ОИЯИ<sup>5/</sup>.

Литература

1. Аниховский В.Е. и др. ОИЯИ, Р11-12809, Дубна, 1980.
2. Руководство по ЕС-1010, том Ш. Система ввода-вывода. VT201.017.12.02-SW. BUDAPEST, 1977.
3. Кнут Д. Искусство программирования для ЭВМ, т.3. М., "МИР", 1978.
4. Хилбурн Дж., Джулич П. Микро-ЭВМ и микропроцессоры. М., "МИР", 1979.
5. Сидоров В.Т., Синаев А.Н., Чурин И.П. ОИЯИ, Р10-12481, Дубна, 1979.
6. Баррон Д. Ассемблеры и загрузчики. М., "МИР", 1974.

Рукопись поступила в издательский отдел  
26 января 1981 года.