

## ТРАНСЛЯТОР С АДРЕСНОГО ЯЗЫКА ДЛЯ ЭВМ «МИНСК-1»

*В. П. Семик, А. Д. Згурский, Л. И. Шкляров, А. Г. Глушко*

В статье рассматривается структура и принцип работы транслятора с входным адресным языком [1] для ЭВМ «Минск-1».

Предлагаемая система автопрограммирования построена таким образом, что ее легко приспособить для ЭВМ других типов, а описание самого транслятора в адресном языке позволяет автоматизировать процесс составления транслятора в кодах конкретной ЭВМ.

Построение транслятора обычно сводится к установлению соответствий между представимыми выражениями входного языка и кодом конкретной машины. Структура транслятора при этом определяется как выбором представимых выражений, так и особенностями языка ЭВМ. В адресном языке можно выделить так называемый машинный стиль, запись алгоритмов в котором отличается от конкретной машинной записи только кодированием. Тогда процесс перевода с входного языка на язык машины можно разделить на два этапа [3]:

1) эквивалентные преобразования во входном языке, переводящие запись алгоритмов в машинный стиль;

2) перекодирование из машинного стиля в код машины.

Если в качестве формул машинного стиля принять канонические формы адресных формул [2], то первый этап не будет зависеть от машинного языка, а сложность второго этапа полностью определится тем, насколько близок язык машины к выбранному стилю. Такое разделение функций транслятора позволяет использовать его для машин других типов, так как в этом случае достаточно заменить только алгоритм перекодирования.

В соответствии с этим транслятор для ЭВМ «Минск-1» состоит из двух основных частей:

1. Блок программирования адресных форм условным кодом (БУК).

2. Блок присвоения истинных кодов (БИК).

Исходной информацией для БУКа служат алгоритмы, записанные во входном языке. В качестве входного языка выбран адресный язык с некоторыми ограничениями, в частности, исключены формулы замены, обмена и относительного перехода.

Формулы языка строятся при помощи адресной формы. Общий вид ее можно описать с помощью металингвистических формул [4] следующим образом:

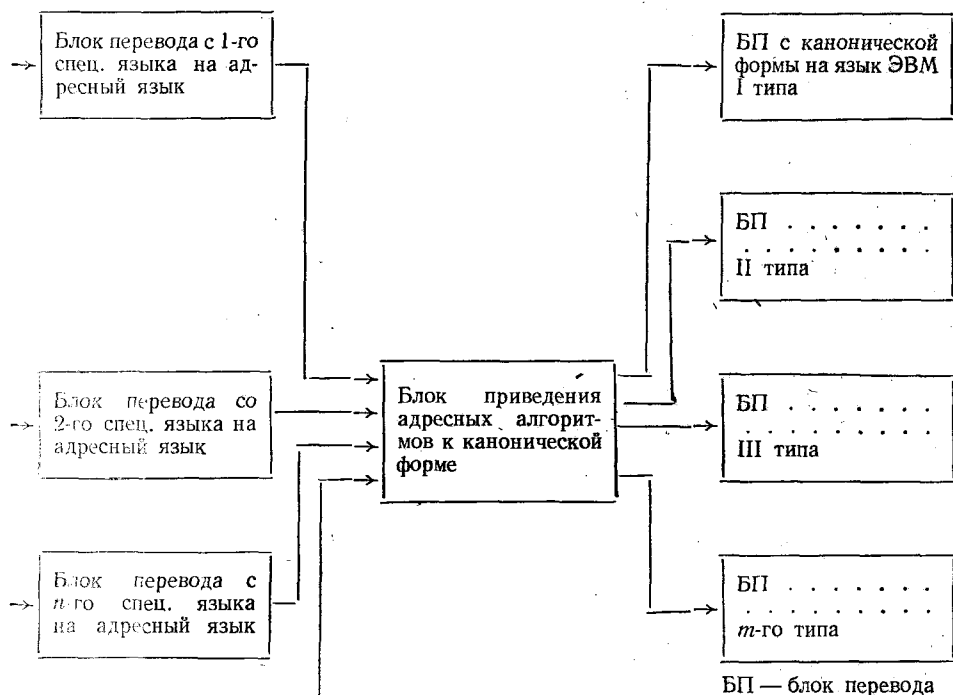
$\langle \text{адресная форма} \rangle ::= \langle \text{адрес} \rangle | \langle \text{метка} \rangle |$   
 $\langle \text{логическое значение} \rangle | O_1 \langle \text{адресная форма} \rangle |$   
 $\langle \langle \text{адресная форма} \rangle O_2 \langle \text{адресная форма} \rangle \rangle.$

Здесь  $O_1$  и  $O_2$  — символы одно- и двуместных операций.

Упрощение алгоритма транслирования достигается за счет унификации объектов языка путем специального кодирования.

Так, например, предикатная формула

$$P\{F_1\}F_2 \downarrow F_3,$$



## СХЕМА

где  $F_1$ ,  $F_2$  и  $F_3$  — адресные формы, сводится по виду к адресной форме, если символы  $P\{, \}$  и  $\downarrow$  формально рассматривать и соответственно кодировать как открывающую скобку (, некоторую двуместную операцию  $O_2$  и совокупность символов закрывающей скобки и той же двуместной операции  $O_2$ . Функции кодирования целесообразно возложить на буквенно-вводное устройство.

Блок вводит информацию поформульно. Формула просматривается один раз поэлементно, слева направо. Элементами считаются скобки, символы одно- и двуместных операций, метки и адреса. Ни одна часть формируемой программы не может считаться законченной, пока не запрограммирована вся формула.

В качестве результата БУК выдает программу в двухадресном стиле канонической формы. Этот же блок производит предварительное распределение памяти.

Результат работы БУКа служит исходной информацией для БИКа. БИК производит окончательное распределение памяти и выдает рабочую программу.

В силу ограниченного объема памяти машины (емкость оперативного запоминающего устройства —  $1024_{10}$  ячеек) и относительную бедность системы команд (отсутствие операций по адресу второго ранга или над индекс-регистрами, операций циклирования, операций в плавающем режиме и т. п.), было выбрано компромиссное решение — разработана система компилирующе-интерпретирующего типа: рабочая программа формируется как в истинном коде машины (фиксированный режим), так и в псевдокоде (плавающий режим). Псевдокод расшифровывается интерпретирующей системой в процессе выполнения рабочей программы. Операция

по адресам высших рангов, вычисляемый переход, операции отношения и одностепенные операции моделируются специальными подпрограммами, а в рабочей программе пишется только обращение к ним.

При необходимости использования специализированных языков эта система может быть полностью использована, а ее входной язык будет служить в качестве языка-посредника.

Общая схема системы автопрограммирования с  $n$  входными специализированными языками для  $m$  различных типов ЭВМ имеет следующий вид, показанный на схеме.

Автоматическое составление транслятора в кодах конкретной ЭВМ осуществляется следующим образом. На заданной машине реализуется универсальный адресный алгоритм [2] — программа, моделирующая машину, языком для которой служит адресный язык, с набором операций, достаточным для работы транслятора. Адресный алгоритм транслятора задается одновременно в качестве программы для универсального алгоритма. В результате получаем программу транслятора в коде заданной ЭВМ.

Авторы приносят благодарность Е. Л. Ющенко за ряд ценных советов, данных им при разработке настоящей системы.

#### ЛИТЕРАТУРА

1. Е. Л. Ющенко. Адресное программирование. Гостехиздат, 1963.
2. В. П. Семик. К вопросу минимизации адресных алгоритмов. «Кибернетика и техника вычислений», Сб. трудов ИК АН УССР. К., 1964.
3. В. П. Семик. До питання алгоритмізації побудови програмуючих програм. «Обчислювальна математика і техніка», Зб. праць ІК АН УРСР, 1963, № 1.
4. Сообщение об алгоритмическом языке АЛГОЛ-60. Изд-во ВЦ АН СССР, М., 1960.