

ооо
оуан

ИНСТИТУТ ТОЧНОЙ МЕХАНИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ АН СССР

Отдел научно-технической информации

62-6
9864-2

Ш

ПРЕДВАРИТЕЛЬНОЕ ОПИСАНИЕ ВЫЧИСЛИТЕЛЬНОЙ
МАШИНЫ "АТЛАС"
ФИРМЫ ФЕРРАНТИ И МАНЧЕСТЕРСКОГО УНИВЕРСИТЕТА

Часть вторая

Математическое описание машины "Атлас"

Москва - 1962

Раздел I - ОБЩЕЕ ОПИСАНИЕ

I.1. СТРУКТУРА СЛОВА

Каждое слово состоит из 48 двоичных разрядов. Команды и числа с плавающей запятой представляются 48-ми разрядными словами. Слово может быть подразделено на два 24-х разрядных полуслова или восемь 6-ти разрядных знаков. Как правило, 3 самых младших разряда модифицированного адреса не учитываются при обращении к слову в запоминающем устройстве (2 самых младших разряда для полуслова), а для некоторых подпрограмм экстракода возможно обращение к отдельным знакам в слове или полуслове.

Можно считать таким образом, что регистр состоит либо:

(1) из 48-ми разрядного слова с адресом r (или $r.0$), либо (2) из 2-х 24-х разрядных слов с адресами r (или $r.0$) и $r.4$, либо (3) из восьми 6-ти разрядных знаков с адресами r (или $r.0$); $r1, r2, \dots, r7$.

(При установлении адреса можно будет понять из контекста, производится ли обращение к слову, полуслову или знаку).

Разряды в команде распределены следующим образом:

$F \quad Va \quad Vm \quad S,$

где:

1. 24 самых младших разряда, т.е. знаки 4,5,6,7(S), как правило, относятся к адресу основного ЗУ.

2. Следующие 7 двоичных разрядов (Vm) относятся к одному из 128 24-х разрядных регистров В (некоторые из этих регистров, используемые для специальных целей, не содержат всех 24-х разрядов. См. раздел 2.2.2).

Кроме случая, где $Vm = 0$, адрес S перед выполнением команды модифицируется содержимым указанного В-регистра.

Однако невозможно В - модифицировать все В - тестовые коды, поскольку в некоторых из них разряды Vm используются как компонента операции.

3. Следующие 7 двоичных разрядов (Va) относятся к одному из этих В-регистров. При одних операциях (коды А) производится вторая В - модификация; при других (коды В) Va определяет тот регистр В, над которым производится операция.

Случай $Va = 122$ является исключением. Эта операция осуществляется не с В122, а с В-регистром, адрес которого хранится в В121. Таким образом, для кода А, S модифицируется при помощи Vm и при помощи содержимого В-ре-



Главное ЗУ может состоять только из ЗУ на сердечниках или ЗУ на сердечниках и барабанах. Однако благодаря встроенной системе и программе постоянного ЗУ, программист может использовать это ЗУ как ЗУ одного уровня.

(а) ЗУ на сердечниках

Данное ЗУ на ферритовых сердечниках имеет время цикла не больше 2 мксек. Это ЗУ состоит из модулей, соединенных попарно, причем емкость модуля - 4 096 слов. Возможно также ЗУ на сердечниках из модулей емкостью 1024 слова, но в дальнейшем будет предполагаться, что емкость модуля 4096 слов. Минимальная емкость ЗУ на сердечниках это есть одна пара модулей, т.е. 8192 слова, и она может быть увеличена в целое число раз. Видимо емкость этой памяти ограничивается практически соображениями, хотя, теоретически, пределом является 2²⁰ слов.

Будет разработан механизм выборки для каждого модуля, причем каждая пара модулей, очевидно, будет состоять из четного и нечетного модуля. Четный модуль будет содержать все регистры с четными адресами для данной пары модулей, а нечетный модуль будет содержать регистры с нечетными адресами. Устройство машины с двумя парами модулей может быть представлено такой диаграммой:

Первая пара модулей
Четный модуль Нечетный модуль

Страница 0 Четные адреса	Страница 0 Нечетные адреса	512 слов на странице
Страница 1 Четные адреса	Страница 1 Нечетные адреса	
Страница 15 Четные адреса	Страница 15 Нечетные адреса	

4096 слов
в модуле

4096 слов

Вторая пара модулей
Четный модуль Нечетный модуль

Страница 16 Четные адреса	Страница 16 Нечетные адреса	512 слов на странице
Страница 17 Четные адреса	Страница 17 Нечетные адреса	
Страница 31 Четные адреса	Страница 31 Нечетные адреса	

4096 слов

4096 слов

При выполнении программы, команды берутся из ЗУ на

сердечниках по парам. Пара команд состоит из одной команды из четного модуля и следующей команды из нечетного модуля той же самой пары модулей. В машине предусмотрены следующие случаи:

(1) Когда первая команда в паре включает передачу управления.

(2) Когда предусмотрена передача управления по второй команде пары.

Машина не имеет специального приспособления для выявления тех случаев, когда вторая команда в паре изменяется первой командой.

(б) ЗУ на магнитных барабанах

На каждом барабане имеется 192 дорожки информации, каждая в 6144 разряда, т.е. всего 24576 слов. Эти дорожки делятся на 8 групп по 24 дорожки, и 24 разряда каждого полуслова хранятся в соответствующих местах на 24 различных дорожках. Передача осуществляется блоками в 512 слов и каждая группа дорожек имеет до 6 таких блоков или секторов. Барабан вращается со скоростью около 5000 оборотов в минуту, так что время оборота составляет 12 мсек, а время передачи одного блока - 2 миллисекунды, без времени ожидания. Передачи с барабана и на барабан замедляют работу вычислительной машины, если обращение производится к той части ЗУ, которая используется при передаче.

(в) Внешнее ЗУ на барабанах

Рассматривается использование барабанов большой емкости типа недавно разработанного фирмой "International Computers and Tabulators". Каждое внешнее ЗУ на барабанах будет иметь емкость около 3x10⁵ слов. Передачи между ЗУ на сердечниках и внешним ЗУ на барабанах могут осуществляться таким же образом, как и передачи между ЗУ на сердечниках и магнитными барабанами. Блок, возможно, будет состоять из 4096 слов.

1.3. ОБОЗНАЧЕНИЕ ТИПА ЗУ

Разряды адреса

23 22 21 20

0 0 0 0 Главное ЗУ (ЗУ на сердечниках и барабанах)

1 0 0 0 Постоянное ЗУ

1 0 1 0

1 1 0 0 ЗУ V

1 1 1 0 Дополнительное ЗУ

} Частное
ЗУ

Автоматическое прерывание (см. раздел 14) имеет место, если:

(а) Обращение к частному ЗУ происходит при работе на Главном Управлении.

(б) Модифицированный адрес находится в частном ЗУ, а немодифицированный адрес не находится в нем при работе на экстракоде или при прерывании. Информация в ячейках постоянного ЗУ может считываться при помощи программы, но не может записываться. Если делается попытка записать в постоянное ЗУ, команда "выполняется" обычным путем, но информация постоянного ЗУ не изменяется.

Раздел 2.- КОД ОПЕРАЦИЙ

2.1.1. РАЗРЯДЫ КОДА ОПЕРАЦИЙ

Десять разрядов для кода операции обозначают следующее:

47	46	45	44	43	42	41	40	39	38	
0	0	0	0	δ	δ	δ	δ	δ	δ	ничего не обозначают
0	0	0	I							Основные коды В
0	0	I	0							Основные тестовые коды
0	0	I	I							Основные коды А
0	I	0	0							Ничего не обозначают
0	I	0	I							Основные коды В + возврат экстракода
0	I	I	0							Основные тестовые коды
0	I	I	I							Основные коды А + возврат экстракода
I	0	δ	δ							Ввод экстракода В
I	I	0	0							Ввод экстракода В
I	I	0	I							Ввод экстракода А
I	I	I	δ							Ввод экстракода А

Коды 0000 или 0100 в разрядах 47-44 ничего не обозначают. Если делается попытка выполнить команду при одном из этих кодов операций, то устанавливается разряд 24 строки I ЗУ V центральной вычислительной машины (см.раздел I3.2) и имеет место автоматическое прерывание.

Если декодируется код 0101 или 0111 в разрядах 47-44, в то время, как работа идет на главном управлении, то выполняется соответствующая основная операция с цифрой 0 в 46-м разряде. Коды с 0110 в разрядах 47-44 являются дубликатами основных тестовых кодов, то есть 0010 в разрядах 47-44. В командах с кодами 0010 или 0110 в разрядах 47-44 разряд 43 не декодируется и, таким образом, имеются два кода для каждой из этих операций; один с нулей в 43-ем разряде, а другой - с единицей.

Вышеуказанное обозначение кодов ввода экстракода допускает 320 экстракодов типа В и 192 экстракода типа А

2.1.2. ЭКСТРАКОД

Команды делятся на две категории;

- 1) основные
- 2) экстракод.

Команды экстракода выполняются при помощи подпрограмм постоянного ЗУ, которые вводятся автоматически, когда старший разряд кода операции в команде есть 1. Они подразделяются на два типа А или В; и соответственно этому имеет место двойная или одинарная модификация адреса. Для экстракодов используются всего 512 кодов, причем из них 192 принадлежат к типу А и 320 к типу В.

В настоящее время первые 2048 слов постоянного ЗУ предназначаются для подпрограмм экстракода. Экстракоды делятся на группы по 64 и каждая группа располагается колонкой по 256 слов в постоянном ЗУ. Поскольку переключение между различными колонками постоянного ЗУ занимает около 1 мксек, желательно, особенно для коротких экстракодов, чтобы все команды для какого-либо данного экстракода находились в одной и той же колонке. Однако, если это невозможно, могут использоваться обыкновенные передачи управления в другую колонку.

Один экстракод может использоваться в качестве подпрограммы другой экстракод.

Обычно имеют место два случая:

а) обычный ввод в подпрограммы и вывод из них при помощи команд возврата. Если имеются только два вывода, то вместо команд возврата могут быть использованы маркеры (т.е. разряд в В-регистре).

Особым является случай, когда команда возврата является общей, при этом нет необходимости в проверке различных ведущих программ и может быть использован обычный вывод экстракода.

б) Необходимо, чтобы подпрограмма, которая является полным экстракодом, выполнялась настолько быстро. Используются 3 метода, каждый из которых слегка удлиняет экстракод ведущей программы, но не влияет на экстракод подпрограммы:

1) Если ведущая программа требует сложных условий для первоначального установления, но имеет то же самое окончание, что и подпрограмма, ввод может быть произведен командой экстракода, чтобы сэкономить место. Вывод осуществляется путем возврата к основной программе;

2) Если конечные условия не одинаковые, необходимо сначала записать содержимое регистра В I27, а затем поместить в В I27 новый адрес постоянного (в)ЗУ, перед тем как ввести подпрограмму экстракода. Подпрограмма экстракода возвращается на главное управление и то, что осталось невыполненным в ведущем экстракоде, выполняется на главном управлении. Окончательный вывод в основную программу происходит путем восстановления В I27.

3) Подобно 2), но подпрограмма возврата требует обращения к частному ЗУ, и поэтому не может быть достигнута на главном управлении. Даются частные команды экстракода, которые передают управление экстракоду, чтобы удовлетворялся ряд условий. В данном случае условия состоят в том, что главное управление содержит адрес с ненулевым знаковым разрядом или в том, что установлен переключатель дополнительного ЗУ. (Переключатель должен быть установлен перед вводом подпрограммы).

Ввод в экстракоды типа А

Почти все вводы в программы экстракода имеют место, когда машина работает на главном управлении.

Случайные вводы могут быть:

а) Когда работа производится на управлении экстракода и второй экстракод используется в качестве подпрограммы.

б) Когда работа производится на управлении экстракода или управления прерывания и ошибка машины (или программа постоянного ЗУ) определяет ввод экстракода.

(а) Ввод при работе на главном управлении

Условия для ввода $f_9 = 1$, $f_8 = 1$ и f_7, f_6 или $0, 1$, или 10 или 11 , где $f_9, f_8, f_7, f_6, f_5, f_4, f_3, f_2, f_1, f_0$ являются разрядами кода операции $I/ME=ME$ и $M/E = M$.

1) $(s + bm + ba)$ хранится в ВII9; то, необходимый адрес может быть определен путем использования ВII9 в качестве модификатора в программе экстракода.

2) Содержимое счетчика команд главного управления (ВI27) увеличивается на 1, т.е. получается адрес следующей команды.

3) Va , т.е. число, определяемое 7-ю разрядами Va , хранится в разрядах 8-2 в ВI2I.

4) Разряды кода операций размещены при управлении экстракода (ВI26) следующим образом:

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4
I	0	0	0	0	0	0	0	0	0	f_8	f_7	f_6	0	0	f_5	f_4	f_3	f_2	f_1
3	2	1	0																
f_0	0	0	0																

5) Разряд M/E вновь устанавливается в Е, чтобы определить управление экстракода.

Цель этого - передать управление соответствующей строке (указанной F) в таблице команд передачи управления в постоянном ЗУ.

(б) Ввод при работе на управлении
экстракода

Условия для ввода такие же, как указывалось выше,
за исключением того, что $M/E = E$.

В результате:

- 1) $(s+bm+ba)$ хранится в В119
- 2) $VI26' = VI26 + I$ (последовательно переписывается)
- 3) $VI21' = Va$
- 4) $VI26' = F$ и это вызывает передачу управления подпрограмме экстракода

(в) Ввод при работе на управлении
прерывания

Если $I/ME = I$, тогда

- 1) $(s + bm+ba)$ хранится в О119
- 2) $VI25' = VI25 + I$
- 3) $VI21' = Va$
- 4) $VI26' = F$
- 5) $M/E = E$

Однако, поскольку управление прерывания всегда пред-
почитается главному управлению и управлению экстрако-
да, немедленный ввод в подпрограмму экстракода не име-
ет места. Когда подпрограмма прерывания выполнена и
 I/ME снова устанавливается в ME , включается управление
экстракода и вводится определенная программа.

Вход в экстракоды типа В

Условиями для ввода являются $f_9 = I$, $f_8 = 0$ или
 $f_8 = I$ и $f_7 f_6 = 00$.

- 1) $(s + bm)$ хранится в В119.
- 2) Число в регистре действующего управления увели-
чивается на 1.
- 3) Va , т.е. число, определяемое 7-ю разрядами Va ,
хранится в В121.

За счет использования В122 (т.е. В-регистра замены)
в подпрограмме экстракода, регистр В, определяемый раз-
рядами Va , может использоваться в качестве второй ком-
поненты операции; а при определении В121 в качестве
компоненты операции могут быть использованы 7 разря-
дов Va .

- 4) $VI26' = F$
- 5) Триггер M/E снова устанавливается в E .

Выход из экстракодов

Если основная команда ($f_9 = 0$) декодируется с
 $f_8 = I$, в то время как соответствующая основная опе-
рация выполняется на управлении экстракода, и если она
является операцией типа В или типа А, тогда ($f_6 = I$).
Затем выполняется команда, непосредственно следующая
за командой, определяющей ввод в подпрограмму экстра-
кода. Если соответствующая основная операция является
тестовым кодом, т.е. в разрядах 45 и 44 стоит 10, тог-
да выполняется основной тестовый код, а триггер M/E
остается в состоянии E . Прерывание по непредусмотрен-
ной операции имеет место в том случае, когда делается
попытка выполнить код, где цифры разрядов 44 и 45 есть
00. При этом триггер M/E остается в состоянии E .

Благодаря возможности прерывания по неэквивалентно-
сти, последняя команда в подпрограмме экстракода не
должна быть командой перехода к адресу в главном ЗУ.

Безусловное нарушение и
непредусмотренные экстракоды

Когда работа производится на главном управлении,
любое обращение к "частному ЗУ", т.е. ЗУ V и дополни-
тельному ЗУ, вызывает автоматическое прерывание (безу-
словное нарушение). При работе на управлении экстрако-
да (а также прерывания) автоматическое прерывание име-
ет место, если модифицированный адрес находится в
частном ЗУ, а немодифицированный адрес там не находит-
ся.

Это не позволяет программисту обращаться непосред-
ственно к ячейкам частного ЗУ.

Не все 64 имеющихся кода в каждой группе распреде-
лены и, следовательно, соответствующие передачи управ-
ления не являются обязательными; таким образом, оста-
ется дополнительное место для необходимых команд в
начале колонки. Там, где это осуществляется, следят
за тем, что если определяется непредусмотренный экстра-
код, и ввод производится частично через предусмотренный
экстракод, выполняемые команды не изменяют соответствую-
щую часть частного ЗУ (кроме, вероятно, общих рабочих
ячеек, относящихся к экстракоду). Таким образом, выпол-
нение непредусмотренного экстракода не помешает работе
какого-либо внешнего устройства или выполнению ка-
кой-либо другой подпрограммы.

2.1.3. РЕГИСТРЫ УПРАВЛЕНИЯ И ТРИГГЕР ПАРЫ

Регистры управления

Регистрами управления являются быстродействующие триггерные регистры, используемые для хранения адресов команд, которые должны выполняться. Когда команда получена по указанному адресу, содержимое используемого регистра управления увеличивается на 1, для того чтобы получить адрес следующей команды в программе. Если команда предназначена для передачи управления, то адрес, указанный в команде, помещается в регистр управления и следующая команда берется по этому адресу.

Организующие подпрограммы и подпрограммы экстракода для машины "Атлас" находятся в постоянном ЗУ. Чтобы обеспечить хранение, а затем восстановление содержимого регистра управления главной программы (В I27), когда должна быть выполнена подпрограмма из постоянного ЗУ, предусмотрены два других регистра управления: - регистр управления экстракода (В I26) и регистр управления прерывания (В I25).

Имеются два "триггера управления" (строка 3, разряд 25 и строка 4, разряд 24 ЗУ Центральной Вычислительной машины - см. раздел I3.2) для определения используемого регистра управления:

триггер "главная программа/экстракод" (М/Е);
триггер "прерывание/главная программа или экстракод" (I/ME).

Выходы этих 2 триггеров определяют, какой из регистров управления используется, а в случае прерывания - какое управление использовалось до этого прерывания.

Возможны 4 случая :

М/Е	I/ME	
0	0	управление экстракода
0	1	управление прерывания (прерывание происходит во время работы на управлении экстракода)
1	0	главное управление
1	1	управление прерывания (прерывание происходит во время работы на главном управлении)

Оба эти триггера можно установить в то или иное состояние программой из постоянного ЗУ, записывая 1 или 0 в соответствующий разряд ЗУ Центральной вычислительной машины.

Управление меняется автоматически:

а) на управление экстракода, если встречается команда экстракода при работе на главном управлении

- б) на главное управление, если встречается команда "возврата с экстракода" (раздел 2.1.2.)
в) на управление прерывания, если встречается "автоматическое прерывание" (см. раздел I4).

Вывод из управления прерывания можно осуществить программой постоянного ЗУ путем вторичного установления разряда I/ME в ME (в ЗУ V).

Триггер пары

Команды из главного ЗУ на сердечниках всегда берутся парами, причем каждая пара включает "четную" команду и "нечетную" команду со следующим (более старшим) адресом. Они помещаются в два триггерных регистра: регистр четной текущей команды (PIE) и регистр нечетной текущей команды (PIO).

С этими регистрами связан триггер "пары". Он устанавливается на "парность", когда используется главное управление и следующая команда, запрашиваемая из главного ЗУ на сердечниках, имеет четный адрес.

Он устанавливается на "непарность":

а) когда при работе или на управлении прерывания следующая команда, запрашиваемая из главного ЗУ на сердечниках, имеет четный адрес;

б) при работе на главном управлении.

Когда команда запрашивается по нечетному адресу из главного ЗУ на сердечниках, она передается прямо из регистра нечетной текущей команды (PIO) в регистр четной текущей команды (PIE) и выполняется, если триггер пары установлен в положение "парность". Если триггер пары не установлен в положение "парность", две команды сначала считываются с главного ЗУ на сердечниках в PIO и PIE и затем необходимая команда передается из PIO в PIE и выполняется.

В случае передачи главного управления на четный адрес, триггер устанавливается в состояние "непарность" (поскольку была приписана М) и затем устанавливается на "парность" после того, как адрес был частично декодирован. Для передачи управления на нечетный адрес триггер остается установленным на "непарность" до тех пор, пока следующая "четная" команда не будет близка к выполнению. Если эта "нечетная" команда вызывает изменение управления, регистр все еще остается установленным на "непарность" и поэтому для получения следующей команды необходимо обращение к ЗУ на сердечниках.

Команды из постоянного и дополнительного ЗУ считываются по одной в регистр четной текущей команды (PIE). "Парный" триггер и регистр нечетной текущей команды (PIO) остаются неизменными, если нет записи в В I27.

При работе на управлении экстракода и прерывания триггер пары остается установленным на непарность, если из главного ЗУ на сердечниках была считана только четная команда.

Команды последовательно берутся попарно из ЗУ на сердечниках, но во всех случаях необходимая команда передается на P1E (через P1O в случае нечетной команды) и выполняется. Таким образом, если желательнее выполнять команды в главном ЗУ на сердечниках при работе на управлении экстракода (или прерывания), вход должен производиться в команду с четным адресом, так чтобы парный триггер устанавливался снова на непарность, иначе первая выполняемая команда будет братья из регистра P1O (или, иначе, триггер мог бы быть установлен на непарность путем записи в В I27).

Программистам не следует использовать четную команду для изменения нечетной команды той же самой пары, иначе можно получить противоречивые результаты при различных однократных выполнениях программы. Как правило, выполняемая нечетная команда будет уже ожидать в регистре P1O, но если прерывание, включающее перемену программы, встречается сразу же после выполнения четной команды, то при возвращении к первой программе (измененные) нечетные команды будут братья из ЗУ на сердечниках.

2.2.2. СПЕЦИАЛЬНЫЕ В-РЕГИСТРЫ

Ниже приводится список В-регистров, используемых для специальных целей.

В-регистр	Назначение
0	Всегда содержит 0
8I-89	Для библиотеки подпрограммы
90	Уровень подпрограммы
9I-99	Для подпрограмм экстракода
100-110	Для организующей программы (супервизор)
III-118	Для подпрограмм прерывания
119	Содержимое адреса экстракода
120	Всегда содержит нуль
121	Регистр замены (см. раздел I.1) Вах.
122	Не существует. Когда он указывается разрядами Ва, вызывается содержимое В I21.
123	Особый регистр, используемый при обращении к внешним устройствам (см. раздел 5.14)
124	Порядок накапливающего сумматора
125	Управление прерывания
126	Управление экстракода
127	Главное управление

Примечание: В-регистры I20-I27 являются триггерными регистрами, а не регистрами ЗУ на сердечниках.

В I24 состоит только из 9 разрядов; 8-разрядный порядок числа хранится в разрядах 22-15, а разряд 23 используется в качестве дополнительного разряда. В разрядах от I4 до 0 - всегда нули. В I21 состоит только из 7 разрядов с 8 по 2; разряды с 23 по 9, I и 0 всегда нули.

2.2.3. РАЗРЯД В-ПЕРЕНОСА

Разряд "В-переноса" это 24-й разряд, строки 6 ЗУ Центральной Вычислительной машины (см. раздел I3.2). Он записывает "перенос или заем" из последнего разряда В-сумматора в ходе В-суммирования некоторых В-команд. Правила для его установки:

1) при работе на главном управлении или управлении экстракода, он устанавливается на I или 0 для В-кодов с разрядами кода операции;

2) при работе на управлении прерывания он специально не устанавливается для любых кодов, а остается таким, каким был установлен ранее;

3) для всех других основных кодов он не устанавливается, а остается прежним.

2.3.1. ПЕРЕЧЕНЬ ОПЕРАЦИЙ

Перечень основных В-операций

	0	I	2	3	4*	5	6	7
O10	$ba' = s - ba$	$ba' = s$	$ba' = ba - s$	$ba' = -s$	$ba' = ba + s$	$ba' = 2^b ba + s$ ЦИКЛИЧЕСКИЙ СДВИГ	$b'a = ba \neq s$	$ba' = ba \& s$
O11	$s' = s - ba$	$s' = -ba$	$s' = ba - s$	$s' = ba$	$s' = ba + s$	$(s' = ba + s)$	$s' = ba \neq s$	$s' = ba \& s$
O12	$ba' = n - ba$	$ba' = n$	$ba' = ba - n$	$ba' = -n$	$ba' = ba + n$	$ba' = 2^b ba + n$ ЦИКЛИЧЕСКИЙ СДВИГ	$ba' = ba \neq n$	$ba' = ba \& n$
O13	код типа n	код типа n	код типа n	код типа n	код типа n	код типа n	код типа n	код типа n
O14	$(ba' = s - ba)$	$(ba' = s)$	$(ba' = ba - s)$	$ba' = ba/2 - s$ ЦИКЛИЧЕСКИЙ СДВИГ	$(ba' = ba + s)$	$(ba' = s)$	$(ba' = ba \vee s)$	$ba' = ba \vee s$
O15	$bt' = s - ba$	код типа s	$bt' = ba - s$	код типа s	код типа s	код типа s	код типа s	код типа s
O16	$(ba' = n - ba)$	$(ba' = n)$	$(ba' = ba - n)$	ЦИКЛИЧЕСКИЙ СДВИГ $ba' = ba/2 - n$	$ba' = ba + (bm \& n)$	$ba' = bm + n$	$(ba' = ba \sqrt{n})$	$ba' = ba \vee n$
O17	$bt' = n - ba$	код типа n	$bt' = ba - n$	код типа n	код типа n	код типа n	код типа n	код типа n

* Перенос устанавливается операцией в колонках, обозначенных звездочкой. Команды, заключенные в скобки, являются дубликатами команд, встречающихся в других местах таблиц. Коды, обозначенные "код типа S", могут вызывать или "неэквивалентность", или прерывание "безусловного нарушения".

2.3.2. ПЕРЕЧЕНЬ ОСНОВНЫХ ТЕСТОВЫХ ОПЕРАЦИЙ

	0	I	2	3	4	5	6	7
O20 (024)	$ba' = n$ $bm' = bm + 0, 04$ если $bm \neq 0$	$ba' = n$ $bm' = bm + I, 0$ если $bm \neq 0$	$ba' = n$ $bm' = bm - 0, 4$ если $bm \neq 0$	$ba' = n$ $bm' = bm - I, 0$ если $bm \neq 0$	$(ba' = n,$ $bm' = bm + 0, 4$ если $bm \neq 0)$	$(ba' = n,$ $bm' = bm + I, 0$ если $bm \neq 0)$	$(ba' = n$ $bm' = bm - 0, 4$ если $bm \neq 0)$	$(ba' = n$ $bm' = bm - I, 0$ если $bm \neq 0)$
O21 (025)	$ba' = n$ если bm не четное	$ba' = n$ если bm четное	$(ba' = n,$ если bm нечетное)	$(ba' = n,$ если bm четное)	$ba' = n,$ если $bm = 0$	$ba' = n,$ если $bm \neq 0$	$ba' = n,$ если $bm \geq 0$	$ba' = n,$ если $bm < 0$
O22 (026)	$ba' = n$ $bm' = bm + 0, 4$ если $bt \neq 0$	$ba' = n$ $bm' = bm + I, 0$ если $bt \neq 0$	$ba' = n$ $bm' = bm - 0, 4$ если $bt \neq 0$	$ba' = n$ $bm' = bm - I, 0$ если $bt \neq 0$	$ba' = n$ если $bt \neq 0$	$ba' = n$ если $bt = 0$	$ba' = n$ если $bt \geq 0$	$ba' = n$ если $bt < 0$
O23 (027)	$(ba' = n$ если $xa = 0)$	$(ba' = n$ если $xa \neq 0)$	$(ba' = n$ если $xa \geq 0)$	$(ba' = n$ если $xa < 0)$	$ba' = n$ если $xa = 0$	$ba' = n$ если $xa \neq 0$	$ba' = n$ если $xa \geq 0$	$ba' = n$ если $xa < 0$

Поскольку в тестовых кодах разряд I9 не декодируется, то для каждой операции (напр. O200 и O240) имеются дубликаты кодов. В операциях 200-208 и 220-223 дополнение прибавляется к bm или вычитается из bm , только в том случае, если проверка прошла успешно. Только в операциях 224-227 и 230-237 имеет место обчная В-модификация n .

2.3.3. ПЕРЕЧЕНЬ ОСНОВНЫХ ОПЕРАЦИЙ НАКАПЛИВАЮЩЕГО СУММАТОРА

	0	I	2	3	4	5	6	7
030	a' = am+s Q EO	a' = am-s Q EO	a' = -am+s Q EO	(a' = -am+s) Q EO	am' = s, l' = 0 Q EO	am' = -s, l' = 0 Q EO	am' = s, l' = 0 Q EO	(am' = s, l' = 0) Q EO
031	a' = am+s N Q EO	a' = am-s N Q EO	(a' = -am+s) Q EO	(a' = -am+s) Q EO	am' = s N AO	am' = -s N AO	(am' = s, l' = 0) Q EO	(am' = s, l' = 0) Q EO
032	am' = am+s QR EO	am' = am-s QR EO	am' = -am+s QR EO	(am' = -am+s) QR EO	(am' = s, l' = 0) Q EO	(am' = -s, l' = 0) Q EO	(am' = s, l' = 0) Q EO	(am' = s, l' = 0) Q EO
033	a' = am+s AO	a' = am-s AO	a' = -am+s AO	(a' = -am+s) AO	am' = s, l' = 0 AO	am' = -s, l' = 0 AO	(am' = s, l' = 0) Q EO	(am' = s, l' = 0) Q EO
034	a' = a Q EO	a' = a EO	a' = am.s Q EO	a' = -am.s Q EO	l' = xs, ya' = ya	l' = xs, m' = sign xs, ya' = ya	s' = am, a' = 0	s' = al, l' = 0
035	(a' = a) Q EO	(a' = a) EO	a' = am.s sign l' = sign m' AO EO	a' = -am.s sign l' = sign m' AO EO	am' = a R AO	s' = am, am' = I	s' = am	s' = al
036	am' = a QR EO	am' = a R EO	am' = am.s QR EO	am' = -am.s QR EO	xa' = xa.8 ya' = ya AO	xa' = xa.8 - I ya' = ya AO	am' = am l' = 0 Q EO	am' = s l' = 0 Q EO
037	(a' = a) Q EO	(a' = a) EO	a' = am.s AO EO	a' = -am.s AO EO	am' = am+s DO QR EO	al' = a+ s m' = rem a+ s DO not set, EO	al' = a+ s m' = rem a+ s DO EO	al' = am+ s m' = rem am+ s DO EO

N в командах EIO, ZII, ZI4 и ZI5 означает, что перед сложением L не считается. Коды, где S не обозначается особо, могут вызвать или "неэквивалентность" или прерывание "безусловного нарушения".

2.4. ПОДПРОГРАММЫ ЭКСТРАКОДА

2.4.1. РАСПРЕДЕЛЕНИЕ НОМЕРОВ ОПЕРАЦИЙ

320 экстракодов типа B (I000-I477) и 192 экстракода типа A (I500-I777) были еще подразделены, как указано ниже. В следующем параграфе показано, как последние два восьмеричных разряда кода операции, где это возможно, соответствуют тем же разрядам подобных основных операций.

I000 - I037	Магнитные ленты.
I040 - I047	Бараны.
I050 - I077	Медленнодействующие внешние устройства.
II00 - II77	Организирующие подпрограммы.
I200 - I237	Тестовые команды.
I240 - I247	Обработка знаков.
I250 - I277	Косвенная адресация.
I300 - I377	Операции B-регистра, преобразования со смешанным основанием.
I400 - I477	Сложные арифметические, векторные и другие операции накапливающего сумматора типа B.
I500 - I477	Арифметические операции с удвоенным количеством разряда и операции, накапливающего сумматора, использующие адрес в качестве компонента операции.
I600 - I647	Логические операции накапливающего сумматора (объединение полуслов).
I650 - I677	Тригонометрические подпрограммы.
I700 - I717	Другие подпрограммы, включая логарифмирование, возведение в степень, извлечение квадратного корня вычисления и т.д.
I720 - I777	Прочие арифметические операции.

2.4.2. АРИФМЕТИЧЕСКИЕ ЭКСТРАКОДЫ

Предполагается, что операции накапливающего сумматора с плавающей запятой округляются, если не оговорено противное. Для простоты там, где это нужно, используются следующие сокращения;

- F - плавающая запятая с округлением
- U - плавающая запятая без округления
- X - используется при работе с фиксированной запятой.

Тестовые команды

- I200 $ba' = n$, если A0 установлено; очистить A0
 I201 $ba' = n$, если A0 очищено; очистить A0
 I202 $ba' = n$, если $m \neq 0$, - I.
 I204 $ba' = n$, если $l = 0$
 I205 $ba' = n$, если $l \neq 0$
 I206 $ba' = n$, если $l > 0$
 I207 $ba' = n$, если $l < 0$
 I216 $ba' = n$, если $bm > 0$ (второй разряд BI27 должен быть нулем)
 I217 $ba' = n$, если $bm \leq 0$ (второй разряд BI27 должен быть нулем)
 I224 $ba' = ba + n$, если $cb = I$ ($cb = B$ -перенос)
 I226 $ba' = n$, если $bt > 0$
 I227 $ba' = n$, если $bt \leq 0$
 I230 $c' = c + 2$, если $am = s$, т.е. если $am - s$ (ненормализованные нормы) начинаются с b идентичных разрядов, где $b = 30$, если $Ba = 0$. в противном случае $b = \min(40, Ba)$
 I234 $ba' = n$, если $m = 0$ (второй разряд BI27 должен быть нулем)
 I235 $ba' = n$, если $m \neq 0$ (второй разряд BI27 должен быть нулем)
 I236 $ba' = n$, если $ka > 0$
 I237 $ba' = n$, если $ka < 0$
 I224 $c' = n$, если накапливающий сумматор нормализован, (младший восьмеричный разряд 1 пропадает)
 I725 $l' = n$, если накапливающий сумматор не нормализован (младший восьмеричный разряд 1 пропадает)
 I727 $l'_c = \begin{cases} c + 1, & \text{если } am > s \\ c + 2, & \text{если } am = s \\ c + 3, & \text{если } am < s \end{cases}$
 I737 $c' = n$, если $|am| < s_{c+1}$; в противном случае $c' = c + 2$.

Обработка знаков

В данном контексте S является адресом знака, т.е. участвуют все 24 разряда.

- I240 $ba' = s$ (в разрядах 0 - 5, остальная часть Ba очищается)
 I241 $s' =$ младший знак ba
 I242 $s' = Ba$ (по модулю 64)
 I244 $ba' = 4$ -м знакам, начиная с S .
 I245 обратная I244- й

- I246 передать 4 м знаков из S в Ba; $ba' = ba + m/2$, где $ba^* = m$ (младшие 2 разряда ba не принимаются во внимание).
 I247 передать 4 м знаков из ba в S , $ba' = ba + m/2$, где $ba^* = m$ (младшие 2 разряда ba не принимаются во внимание).

Косвенная адресация

В этих экстракодах компонента операции является содержимым $s + ba$, т.е. предполагаемый адрес модифицируется bm , и полученный таким образом косвенный адрес модифицируется при помощи ba .

- I250 $am' = sX$
 I256 $s' = amX$.

Преобразования со смешанным основанием

- I330. Преобразовать число со смешанным основанием в двоичное целое число, $am' = r_7(r_6(\dots(r_1(r_0 am + S_0) + \dots) + S_7)$, где r_1, s_1 - четыре младших разряда знаков в ba , S_7 соответственно.
 I331. Преобразовать двоичное целое число со знаком в форму со смешанным основанием в знаках, начиная с s . am' - не преобразованная часть am . A0 устанавливается только в том случае, если знак минус не может быть введен правильно, Ba содержит произведение оснований. Ba^* содержит основания и информацию об исключении нулей.
 I332. Преобразовать десятичное число, записанное в ba знаках по 6 двоичных разрядов, начиная со знака, адрес которого S , в восьмеричную форму с плавающей запятой в am . b II9' - адрес следующего знака после обработанного. Десятичное число может быть в форме с фиксированной или плавающей запятой и могут появиться знаки + - . , .
 S_p и NL в начале числа не будут учитываться, а S_p не будет учитываться в начале порядка числа. В противном случае NL и S_p будут рассматриваться как границы числа.
 I333. Преобразовать am в ряд десятичных знаков, начиная со знака, адрес которого S ; знаки + заменяется S_p . Полный диапазон указывается в разрядах 3-8 ba , а число требуемых десятичных знаков в разрядах 9-14 ba . Если указан диапазон недостаточный, десятичный порядок числа выходит на место нескольких последних знаков.
 I334. Преобразовать целое число, хранимое в десятичных знаках в H: в целое число в ba .

- I335. Преобразовать целое число ba в десятичные знаки в Н.
 I336. $ba' = n$, $am' = am \cdot 10^n$, где $\frac{1}{10} \leq am / 10^n < 1$.

Операции В-регистра

В некоторых из этих операций имеет значение положение десятичной запятой. Если не оговорено иное, запятая занимает один восьмеричный разряд со стороны младших разрядов.

- I301 $ba' = -\frac{1}{2}$ целой части от $8a$
 I302 $ba' = ba \cdot n$
 I303 $ba' = -ba \cdot n$
 I304 $ba' =$ целая часть $(ba + n)$, (восьмеричная дробь равна нулю)
 I305 $ba' =$ остаток $(ba + n)$
 I312 $ba' = ba \cdot n$ (для 24-х разрядных целых чисел)
 I313 $-ba' = -ba \cdot n$ (для 24-х разрядных целых чисел)
 I314 $ba' =$ целая часть $(ba + n)$ (для 24-х разрядных целых чисел)
 I315 $ba' =$ остаток $(ba + n)$
 I320 $ba' =$ содержимое I-го разряда в n .
 I321 $ba' =$ положение I-го старшего разряда в n ; если $n=0$, $ba'=24$
 I323 $ba' = ys/2$ в разрядах 0-9, остаток нуль (2-ой разряд $s+bm$ учитывается)
 I324 $ys' = 2ba$ (2-ой разряд $s + bm$ учитывается)
 I325 $ba' = n \cdot 2^{12}$ (установленный порядок)
 I340 арифметический сдвиг вправо без округления
 I341 арифметический сдвиг влево без округления
 I342 циклический сдвиг вправо (n берется по модулю 24)
 I343 циклический сдвиг влево (n берется по модулю 24)
 I344 логический сдвиг вправо
 I345 логический сдвиг влево
 I346 $s' = sv \ ba$
 I356 $bt' = ba \neq s$
 I357 $bt' = ba \ \& \ s$
 I364 $ba' = ba \ \& \ n \ \vee \ bm \ n$ (заменить часть, обозначенную маской (2-ой разряд $BI27$ должен быть равен нулю))
 I376 $ot' = ba \neq n$
 I377 $bt' = ba \ \& \ n$.

Арифметические операции над комплексными числами

Эти операции включают использование комплексного накапливающего сумматора Ca , который представляет собой пару последовательных регистров, первый из которых имеет адрес ba . Если $Ba = 0$, тогда ca занимает ячейки 0,1. Все эти операции используют A в качестве рабочей ячейки. В данном контексте s обозначает число с удвоенным количеством разрядов.

- I400 $ca' = \log s$;
 I401 $ca' =$ возведение в степень s ;
 I403 $ca' =$ сопряж. s ;
 I407 $ca' =$ обратная величина s ;
 I410 $ca' =$ кв. корень из s ;
 I414 $am' =$ модуль s ;
 I415 $am' = \arg. \ s$: (в радианах)
 I416 $ca' = scos \ s^*$, $s \sin \ s^*$
 I420 $ca' = ca + s$;
 I421 $ca' = ca - s$;
 I424 $ca' = s$;
 I425 $ca' = -s$;
 I426 $s' = ca$
 I462 $ca' = ca \cdot s$;

Векторные операции

Порядок векторов выражается n . Вектор s_1 хранится в последовательно расположенных ячейках, начиная с ba ; s_2 хранится подобным же образом в ba^* . Накапливающий сумматор используется в качестве рабочей ячейки.

- I430 $s_1' = s_1 + s_2$
 I431 $s_1' = s_1 - s_2$
 I432 $s_1' = am \cdot s_2$
 I433 $s_1' = s_1 + am \cdot s_2$
 I434 $s_1' = s_2$
 I436 $am' = \sum_{i=0}^{n-1} s_{1i} s_{2i}$
 I437 $a' = \sum_{i=0}^{n-1} s_{1i} s_{2i}$

Смешанные операции накапливающего сумматора типа В

- I452 $m' = m \cdot xs \cdot 8^{ya+ys-ba}$; $ya' = ba \cdot (X)$
 I460 начало процесса получения случайных чисел:
 $am' = s' =$ первое значение
 I461 выработать псевдослучайное число: $am' = s' = f(s)$
 I463 метод Рунге - Кутты - Гилла
 I464 $s_1' = am' = am \cdot s + s_1(F)$, где $S_1 = ba$ (кумулятивное умножение)
 I465 $s_1' = a' = am \cdot s + s_1$; где $S_1 = ba$ (кумулятивное умножение с удвоенным количеством адресов).
 I466 $a' = C(S + bm + ba) \cdot C(S + bm) + a$
 I467 $am' = \sum_{i=0}^r s_1 x^i$ где $S_1 = S+i$, $x=am$, $r=Ba$ (по модулю 64)
 I470 $C(ba)' =$ целая часть $(am+n)$, $am' =$ остаток F
 I471 $C(ba)' =$ целая часть $(a+n)$, $am' =$ остаток X
 I472 $m' = (m+xs) \cdot 8^{ya-ys-ba}$, $ya' = ba$ X

I473	$c(ba)'$	= частное $(am+s)$, am' = остаток	X
I474	$c(ba^*)'$	= целая часть $(am+s)$, am' = остаток	F
I475	$c(ba)'$	= частное $(a+s)$, am' = остаток	X
I476	$c(ba)'$	= частное $(a+s)$, am' = остаток	F
I477	$c(ba)'$	= частное $(am+s)$, am' = остаток	F

Арифметические операции с удвоенным количеством разрядов

Число с удвоенным количеством разрядов формируется при помощи $s: = s + s^*$.

I500	$a' = a + s:$
I501	$a' = a - s:$
I502	$a' = -a + s:$
I504	$a' = s:$
I505	$a' = -s:$
I510	$a' = a + s$
I511	$a' = a - s$
I512	$a' = -a + s$
I542	$a' = a.s:$
I543	$a' = -a.s:$
I556	$s' = a$
I576	$a' = a \div s:$

Арифметические операции, использующие адреса в качестве компоненты операции

Адрес интерпретируется как 21-разрядное целое число с одним восьмеричным дробным разрядом. Поэтому операции с фиксированной запятой предполагают порядок чисел равным 12.

I520	$am' = am + n$	F
I521	$am' = am - n$	F
I522	$am' = -am + n$	F
I523	$am' = -am - n$	F
I524	$am' = n, l' = 0$	F
I525	$am' = -n, l' = 0$	F
I530	$a' = am + n$	X
I531	$a' = am - n$	X
I532	$a' = -am + n$	X
I533	$a' = -am - n$	X
I534	$am' = n, l' = 0$	X
I535	$am' = -n, l' = 0$	X
I552	$al' = am.n$	X
I553	$al' = -am.n$	X
I562	$am' = am.n$	F
I563	$am' = -am.n$	F
I574	$am' = am + n$	F

Операции логического накапливающего сумматора

Логический накапливающий сумматор G в действительности представляет собой B 98, 99

I600	$g' = -g + s$
I601	$g' = s$
I602	$g' = g - s$
I603	$g' = -s$
I604	$g' = g + s$
I605	$g' = g \oplus s$ (циклический перенос)
I606	$g' = g \neq s$
I607	$g' = g \& s$
I613	$s' = g$
I626	$g' = \sim g$
I630	$g' = g \cdot \bar{s}$
I634	$am' = g$
I637	$g' = am$
I642	- циклический сдвиг вправо (n берется по модулю 48)
I643	- циклический сдвиг влево (n берется по модулю 48)
I644	- логический сдвиг вправо
I645	- логический сдвиг влево
I646	$g' = g \vee s$

Объединение полуслов

Эти операции являются операциями с плавающей запятой и с округлением; где это возможно 8-ми разрядный порядок числа сохраняется, но дробная часть уменьшается до 16 разрядов. Предполагается, что накапливающий сумматор содержит стандартное число с плавающей запятой

I616	$h' = am$
I617	$am' = am \div h$
I620	$am' = am + h$
I621	$am' = am - h$
I622	$am' = am.h$
I623	$am' = -am.h$
I624	$am' = h$
I625	$am' = -h$

Вычисление тригонометрических функций

В этих программах аргумент "s" определяется следующим образом:

если $S+bm+ba = 0$, аргумент будет am ; в противном случае, аргумент будет равен $S(S+bm+ba)$.

I650	$am' = \sin "s"$	(в радианах)
I651	$am' = \cos "s"$	(в радианах)
I652	$am' = \text{tg} "s"$	(в радианах)
I660	$am' = \arcsin "s"$	(в радианах)
I661	$am' = \arccos "s"$	(в радианах)
I662	$am' = \text{arctg} "s"$	(в радианах)
I672	$am' = \text{arctg} am/s$	(в радианах)

Другие операции, включая вычисление показательной функции и извлечение квадратного корня

В этом контексте аргумент "s" определяется следующим образом:

Если $S+bm+ba=0$, аргументом является am; в противном случае, аргумент будет выражен $S(S+bm+ba)$.

I700 am' = log "s"
 I701 am' = возведено в степень "s"
 I704 am' = целая часть "s"
 I705 am' = дробная часть "s"
 I706 am' = знак "s" (т.е. = +I, 0, -I в соответствии с тем, "s" >, =(или) < 0)
 I707 am' = обратная величина "s"
 I710 am' = кв.кор. "s" F
 I711 am' = кв.кор. "s" FX
 I714 am' = кв.кор. (am² + s²)

Другие операции накапливающего сумматора

I720 s' = s + am (F)
 I721 s' = s - am (F)
 I722 s' = -s + am (F)
 I730 s' = s + am (X)
 I731 s' = s - am (X)
 I732 s' = -s + am (X)
 I740 al' = al + s, m' = m + переполнение 1'(X)
 I741 al' = al - s, m' = m + переполнение 1'(X)
 I742 al' = s, m' = 1s'(X)
 I743 al' = -s, m' = 1s'(X)
 I744 l' = -xs (X)
 I745 l' = -xs, m' = 1s (X)
 I746 s' = am, am' = s (X)

I752 m' = m.xs.8^{ya+ys-12}; ya' = I2
 I754 округлить a, добавляя I к младшему разряду M, если старший разряд L - единица, затем нормализовать l' = 0.

I760 m' = m.2⁻ⁿ арифметический сдвиг вправо.
 I761 m' = m.2ⁿ арифметический сдвиг влево
 I762 m' = m.8^{ya-n}, ya' = n арифметический сдвиг для выделения порядка числа

I763 xa' = xa.8^{ya-n}, ya' = n арифметический сдвиг для выделения порядка числа

I764 xa' = xa.2⁻ⁿ арифметический сдвиг вправо
 I765 xa' = xa.2ⁿ арифметический сдвиг влево
 I766 am' = |am| (X)
 I767 am' = |s| (X)

I772 m' = (m/xs).8^{ya-ys+12}, ya' = I2
 I775 am' = a + s (X)
 I776 am' = s + am (F)
 I777 am' = s + am (F)

2.4.3. ОРГАНИЗУЮЩИЕ ЭКСТРАКОДЫ

ПРИМЕЧАНИЕ. Этот раздел неполный. Он будет дополнен, когда будут детально разработаны экстракоды для различных программ (напр., организующей, контролирующей, компилирующей и т.п.).

Ввод и вывод подпрограммы

II00 Войти в подпрограммы в S (команда возврата хранится в b90)
 b90' = b90 + I, 0
 II01 Выйти из подпрограммы и пропустить n слов (связь программы возврата хранится в b90)
 b90' = b90 - I, 0.
 II02 Войти в подпрограмму в S ; ba' = c + I.

Команды ветвления

В этом контексте B = Ba (по модулю 64).
 II04 начать условный переход B от S.
 II05 ликвидировать условный переход B (если B = 0, ликвидировать действующий условный переход)
 II06 c' = n, если условный переход B действующий
 II07 назначить демпфирование в S для B условных переходов (приблизительно требуется 20B слов).

Установка ловушки

III0 Установить способ постановки ловушки
 III1 Установить обыкновенный способ
 III2 Ловушка

Другие передачи

II20 ba' = синхронизация
 II21 ba' = данные
 II22 ba' = счетчик команд
 II23 установить счетчик команд = n.2¹⁰
 II24 ba' = v6
 II25 v6' = ba

наковы. Нуль с плавающей запятой представляется как $0 \times 8-128$ и считается нормализованным числом.

Мантисса накапливающего сумматора

Арифметические операции с плавающей запятой использует накапливающий сумматор с плавающей запятой, в котором мантисса имеет двойную длину (знак + 78 разрядов). Чтобы избежать потери информации вследствие "переполнения", когда выполняется команда, предусмотрены два старших дополнительных разряда сумматора. При подаче числа из ячейки ЗУ в старшую половину накапливающего сумматора, знаковый разряд числа также передается в два дополнительных разряда. Передача числа старшей половины накапливающего сумматора в регистр ЗУ означает перенос знакового разряда и 39-ти старших разрядов после знакового разряда.

В дополнение к 79 разрядам в мантиссе накапливающего сумматора имеется еще один разряд, в котором находится знак младшей половины сумматора. Этот разряд устанавливается на нуль при помощи:

а) Большинство команд сложения и вычитания, где L сначала очищается (300-303, 312, 313, 320-323, 330-333).

б) Команд, которые передают информацию из ячейки ЗУ в M или L (включая $am' = -S$) и обратно, и где L или A очищаются особым образом (304-307, 316, 317, 324-327, 334-337, 346, 347).

в) Шести команд умножения (342, 343, 362, 363, 372, 373).

г) Четырех команд деления (374-377).

Разряд остается неизменным, в случаях:

а. двух команд сложения и вычитания, где L первоначально не очищается (310, 311);

б. команд, которые передают число из ячейки ЗУ в M без очищения L (314, 315);

в. команд, определяющих комбинацию округления, нормализации и проверки на переполнение порядка числа, но без выполнения какой-либо арифметической операции (340, 341, 350, 351, 354, 360, 361, 370, 371);

г. команд, которые передают число из M или L в ячейку ЗУ, не очищая A или L (355-357);

д. двух команд, определяющих один восьмеричный сдвиг в содержимом A (364, 365);

е. двух команд взятия по модулю (366, 367).

Разряд устанавливается в 0 или в 1 в зависимости от знака L при помощи:

а. команд, которые передают содержимое ячейки ЗУ в L (344, 345);

б. двух команд умножения без нормализации и округления (352, 353).

Порядок числа в накапливающем сумматоре

Порядок числа в накапливающем сумматоре хранится в разрядах 22-15 регистра В 124; разряд 23 используется

как дополнительный разряд для выявления переполнения или выхода из разрядной сетки порядка числа.

Остающиеся разряды В-124 содержат нуль. Разряд 15 в порядке числа означает 1 и разряд 22 означает 128. Цифра 22-го разряда В-сумматора регистра В-124 также поступает не только в 22-ой разряд регистра В-124, но также и в разряд 23-й. 22-ой разряд регистра В-124 переносится в 22 и 23-й разряд В-сумматора. Таким образом, невозможно поставить дополнительный разряд порядка числа сумматора в условия переполнения или выхода из разрядной сетки при помощи одного из кодов В.

В "Атласе" используется восьмеричный порядок числа и поэтому любой сдвиг мантиссы должен осуществляться на три разряда, чтобы изменить порядок чисел на единицу. Когда имеет место сдвиг вниз, более старший дополнительный разряд (но не знаковый разряд) мантиссы повторяется в новом содержании знакового и дополнительных размеров.

Стандартные числа

В перечне основных кодов многие операции определяются как нормализующие содержимое накапливающего сумматора после операции (например, 0300). В этих случаях содержимое накапливающего сумматора сдвигается вверх или вниз (с соответствующим изменением порядка числа) до тех пор, пока не получится нормализованное число. Сдвиг на 6 двоичных разрядов можно осуществить в то же самое время, что и сдвиг на 3 разряда, т.е., если требуется сдвиг больший, чем на 3 разряда, содержимое A сдвигается на 6 разрядов сразу до тех пор, пока число не нормализуется, или не будет нужен сдвиг только на 3 разряда и тогда этот 3-х разрядный сдвиг осуществляется. Например, если содержимое накапливающего сумматора нужно сдвинуть вверх на 15 разрядов, чтобы получить нормализованную форму, это осуществляется при помощи 2-х сдвигов на 6 разрядов и одного сдвига на 3 разряда. Подобное же разбиение на 6-ти разрядные и 3-х разрядные сдвиги имеет место, когда содержимое накапливающего сумматора или число из ячейки ЗУ должно быть денормализовано для сложения или вычитания с плавающей запятой. Прежде чем производить нормализацию, производится специальная проверка, чтобы выяснить, не содержит ли накапливающий сумматор нуль; если это так, сдвиг не производится и порядок числа меняется на -128 .

Говорят, что содержимое накапливающего сумматора имеет "сверхнормализованную" форму, если знаковый разряд и два дополнительных разряда дробной части не одинаковы. Для ненормализованных кодов (например, 0330) содержимое накапливающего сумматора можно оставить или в сверхнормализованной или недонормализованной форме. В первом случае устанавливается разряд переполнения накапливающего сумматора в ЗУV, но если делается попытка передать содержимое накапливающего сумматора в ЗУ, дополнительные разряды не передаются и поэтому в ЗУ может быть пе-



редан неправильный ответ. Например, если $-I,0$ прибавляется к $-I,0$, при использовании ненормализованного кода (например, 0330) операция имеет вид (при том допущении, что нет необходимости в действии с порядками):

$d_{-2} \ d_{-1} \ d_0 \ d_1 \ d_2 \ d_3 \ \dots \ d_{39} \ d_{40} \ \dots \ d_{78}$
начальное содержимое A

I I I, 0 0 0 . . . 0 0 . . . 0

содержимое S

I I I, 0 0 0 . . . 0

Окончательное содержимое A

I I 0, 0 0 0 . . . 0 0 . . . 0

где $d_{-2} \ d_{-1}$ - дополнительные разряды

d_0 - знаковый разряд

$d_1, d_2 \dots d_{78}$ - разряды накапливающего сумматора.

После того, как передана старшая половина A в строку ЗУ, передаются порядок числа и разряды с d_0 по d_{39} включительно, т.е. число с мантиссой 0.

Подобным же образом, если операции с плавающей запятой выполняются, когда начальное содержимое A является сверхнормализованным, может быть получен неправильный ответ.

Округление

Принятый метод состоит в том, что для операций с плавающей запятой при округлении вводится единица в младший значащий разряд M, если содержимое L не равно нулю. Единица не вводится, если содержимое L - нуль.

Для кодов, с нормализацией и округлением, сначала производится нормализация. Введение единицы в самый младший разряд M, если содержимое L не нуль, не может сделать нормализованное число ненормализованным и, следовательно, дальнейшая нормализация не является необходимой.

Ошибка при этом методе округления:

Младший разряд M (первоначально)	Содержимое (39 разрядов)	Младший разряд M после округления	Ошибка
0	000 00	0	0
0	000 0I	I	+III . . . II
0	000 IO	I	+III . . . IO
0	I00 00	I	+I00 . . . 00
0	III IO	I	+000 . . . IO
0	III II	I	+000 . . . 0I

I	000 00	I	0
I	000 0I	I	-000 . . . 0I
I	I00 00	I	-I00 . . . 00
I	III IO	I	-III . . . IO
I	III II	I	-III . . . II

Из этой таблицы видно, что средняя ошибка (при допущении, что все возможные значения содержимого L равновероятны) при округлении окончательного содержимого накапливающего сумматора двойной длины до результата длиной в одно слово - нуль, и таким образом результат не смещается.

Следующее преимущество состоит в том, что арифметические операции с целыми числами обычной длины могут быть выполнены точно без каких-либо ошибок округления и при этом в каждом случае результат операции находится полностью в M и не распространяется в L.

Команда (0354) предусмотрена для того, чтобы дать где это требуется, тип округления "плюс один", т.е. прибавить к самому младшему разряду M, когда самый старший разряд L есть I.

Таблица, соответствующая вышеприведенному, дает:

Содержание L (39 разрядов)	Прибавляется к M	Ошибка
000 00	0	0
000 0I	0	-000 0I
000 IO	0	-000 IO
0II II	0	-0II II
I00 00	I	+I00 00
I00 0I	I	+0II II
III IO	I	+000 IO
III II	I	+000 0I

Снова, исходя из предположения, что все возможные значения содержимого L почти одинаковы, имеется смещение для того случая, когда L содержит I00 00. (В операциях с десятичными знаками тип округления "плюс один" даст несмещенную ошибку; в том случае, когда нужно округлить . . . 500 . . . 0, единица прибавляется к следующему, непосредственно слева за пятеркой, разряду, если этот разряд нечетный; 5 отбрасывается без добавления I к следующему разряду, если этот разряд четный).

Сложение и вычитание

Прежде чем два числа, представленные в форме с плавающей запятой, могут быть сложены или вычтены, числа должны быть сдвинуты относительно друг друга так, чтобы разряды, имеющие одинаковое значение, складывались вместе. Размер требуемого сдвига определяется разностью порядков двух этих чисел.

В "Атласе" число, имеющее меньший порядок, сдвигается вниз, в A_1 . Если имеет место нечетное число восьмеричных сдвигов, первый сдвиг - единичный, а затем сдвигаются два восьмеричных разряда за один раз. Для округления и операций с удвоенным количеством разрядов необходимо, чтобы окончательное содержимое накапливающего сумматора (перед округлением, если оно производится) всегда имело правильное представление числа двойной длины, и иногда необходимо произвести специальное действие в кодах типа $(am - s)$ и $(-am + s)$ перед сдвигом. Это специальное действие необходимо, поскольку накапливающий сумматор связан только с A_m , а не с A_1 . Если код принадлежит к типу $(am - s)$ и порядок s меньше, тогда x_s снабжается знаком минус, прежде чем произойдет сдвиг и схемные сумматоры меняются для выполнения $(am + s)$.

Подобным же образом, если код имеет вид $(-am + s)$ и am следует сдвинуть, ему сначала дается знак минус и схемы сумматора изменяются для выполнения $(am + s)$.

В то же самое время, когда производится сдвиг, порядок числа накапливающего сумматора устанавливается на больший из двух порядков.

Два числа затем складываются или вычитаются (эта операция занимает более 42 разрядов), затем следует нормализация и округление. Во всех кодах устанавливается разряд E_0 , если это нужно, перед тем, как выдается сигнал окончания работы сумматора.

В случае нормализованных кодов, если накапливающий сумматор содержит нуль при входе в цикл нормализации, порядок числа устанавливается на $-I28$.

Числа с порядками, различающимися на 16 или более в случае кодов с округлением и на 32 или более в случае кодов без округления, находятся "вне диапазона". Числа "вне диапазона" рассматриваются особым образом, поскольку содержимое накапливающего сумматора заменяется числом с большим порядком; это число снабжается знаком "минус", если это необходимо, т.е., если код принадлежит к типу $(am - s)$ и если числа находятся вне диапазона с порядком большим чем s , накапливающий сумматор устанавливается на $-s$. (Числа 16 и 32 были выбраны для определения "вне диапазона", так как они являются первыми ближайшими большими степенями двух, для чисел 13 и 26, соответственно).

Умножение

Стандартный процесс состоит в следующем:

(1) $(y_a + y_s)$ формируется и проверяется на выход из разрядной сетки. Если имеет место выход из разрядной сетки, указанная ниже операция (5) выполняется после того, как обработан первый восьмеричный разряд, и накапливающий сумматор устанавливается на нуль с плавающей запятой.

(2) Знак L и его разряды q очищаются (разряды q представляют собой три разряда от нижнего конца L).

(3) Содержимое строки ZU переписывается в L и знак запоминается (не на месте разряда знака L).

(4) am (включая 2 дополнительных разряда) запоминается в регистре " X_s " в накапливающем сумматоре; Zam также формируется и запоминается в этом сумматоре и am очищается.

(5) Умножение производится по одному восьмеричному разряду целиком и величина, которую следует прибавить к неполному результату, определяется по 3-м нижним разрядам L и одному верхнему разряду q .

(а) Если код принадлежит к типу $(am.s)$, тогда количество, которое добавляется к накапливающему сумматору, помещается в третьей колонке следующей таблицы (где am - первоначальное содержимое сумматора). Соответствующее количество для кода типа $(-am.s)$ дается в 4-ой колонке.

Нижний восьмеричный разряд	Верхний q разряд	$am.s$ код	$-am.s$ код
0	0	0	0
1	0	am	$-am$
2	0	2 am	- 2 am
3	0	3 am	- 3 am
4	0	- 4 am	4 am
5	0	- 3 am	3 am
6	0	- 2 am	2 am
7	0	- am	am
0	I	am	- am
1	I	2 am	- 2 am
2	I	3 am	- 3 am
3	I	4 am	- 4 am
4	I	- 3 am	3 am
5	I	- 2 am	2 am
6	I	- am	am
7	I	0	0

(б) Весь накапливающий сумматор сдвигается вниз на один восьмеричный разряд и производится сложение в следующем восьмеричном разряде.

(в) Регистры накапливающего сумматора располагаются

таким образом, что последний восьмеричный разряд (т.е. I4-ый) является знаковым разрядом строки ЗУ, т.е. он равен нулю, если x_8 положительно, и 7-ми, если x_8 отрицательно.

Вышеуказанный процесс остается одинаковым для всех кодов умножения. После этого предпринимаются следующие действия для отдельных кодов:

- 342-343 Вводится цикл нормализации и после нормализации порядок числа проверяется на переполнение.
- 352-353 Знак L устанавливается равным конечному знаку am , устанавливается разряд переполнения накапливающего сумматора, если это необходимо, и порядок числа проверяется на переполнение.
- 362-363 Вводится цикл нормализации и после нормализации am округляется, если необходимо, и порядок числа проверяется на переполнение.
- 372-373 Устанавливается разряд переполнения накапливающего сумматора, если необходимо, и порядок числа проверяется на переполнение.

Деление

Последовательность во всех операциях следующая:

- 1) Формируется ($ya - ys$) и проверяется на переполнение или на выход из разрядной сетки. Если имеет место переполнение, устанавливается разряд EO в ЗУV Центральной вычислительной машины и осуществляется операция деления. Если имеет место выход из разрядной сетки, накапливающий сумматор очищается до нуля с плавающей запятой и дается команда "наполнение сумматора".
- 2) Очищается знаковый разряд L.
- 3) Сравниваются знаки a и x_8 и знак частного (0 или 1, в соответствии с тем, имеет ли a и x_8 одинаковые или разные знаки) запоминается как "разряд Q_8 " (строка 6, разряд 28 ЗУ V Центральной вычислительной машины).
- 4) Берется модуль x_8 и, если результат сверхнормализован, x_8 сдвигается вниз на один восьмеричный разряд и единица вычитается из порядка числа.
- 5) Формируется модуль am (т.е. $am' = |am|$) и, если результат сверхнормализован, нормализуется весь сумматор.
- 6) ($xa - xs$) формируется в буфере сумматора и, если это число отрицательно, т.е. $x_8 > xa$, то xa сдвигается вверх на один восьмеричный разряд, единица вычитается из порядка числа и "I2/I3 разряд" (строка 6, разряд 29 ЗУ V Центральной вычислительной машины) устанавливается на нуль; если $xa \geq xs$, разряд I2/I3 устанавливается равным единице.
- 7) Начинается цикл деления и повторяется I3 раз. Ответ формируется в трех разрядах "q" (которые находятся в младших разрядах L), и содержимое накапливающего сумматора, включая разряды q, сдвигается вверх на один восьмеричный разряд по мере формирования одного восьмеричного разряда. В итоге, частное находится в L и "ос-

таток" сдвинут вверх, в M.

8) Окончательное действие над всеми кодами перед подачей сигнала наполнения накапливающего сумматора состоит в проверке переполнения порядка числа.

Добавочные действия над отдельными командами:

Команда 374

- (a) x_8 проверяется на нуль или недонормализацию; если это так, происходит прерывание. Устанавливается разряд DO и дается сигнал наполнения накапливающего сумматора; в содержимом M никаких изменений не происходит.
- (б) L первоначально очищается.
- (в) После того, как am и x_8 были сделаны положительными, к порядку числа добавляется I4.
- (г) После цикла деления ответ округляется путем введения единицы в самый младший разряд L, если содержимое M не равно нулю.
- (д) M очищается и накапливающий сумматор может быть нормализован.
- (е) M делается отрицательным, если разряд Q_8 равен единице и накапливающий сумматор снова нормализуется, если это необходимо.

Команда 375

- (a) После выполнения рассмотренного выше пункта (5), единица добавляется к содержимому Ya .
- (б) После цикла деления содержимое M сдвигается вниз на один восьмеричный разряд, причем, переписывается начальное содержимое самого младшего восьмеричного разряда M. Это означает, что, если $x_8 \leq xa$, информация, которая была в самом младшем разряде L перед началом деления и которая была сдвинута вверх в M, теряется. Кроме того, окончательное содержимое M будет иметь неверный знак, если двоичный разряд, следующий за знаковым разрядом, был единицей перед последним восьмеричным сдвигом в цикле деления, т.к., когда M снова сдвигается вниз, этот разряд будет передан в дополнительные разряды и знаковый разряд M.
- (в) Частное находится в L со своим порядком в Ya и остаток (со знаковым разрядом, который должен корректироваться при помощи программы фиксированного ЗУ) находится в M, где его порядок берется равным:

$Ya - I2$, если $|x_8| \leq |xa|$ (в разряде I2/I3 ставится 1)

$Ya - I3$, если $|x_8| > |xa|$ (в разряде I2/I3 ставится 0)

Прерывания не происходит, если x_8 равно нулю или недонормализовано.

Если x_8 - нуль, окончательное содержимое L - одна единица, и окончательное содержимое M есть начальное содержимое самого младшего восьмеричного разряда M и I2-ти старших восьмеричных разрядов L. Окончательно, дополнительные и знаковые разряды M представляют собой то

же самое, что и начальный старший двоичный разряд самого младшего восьмеричного разряда М. Разряд I2/I3 - единица, и разряд Qs - тот же самый, что и прежний знаковый разряд М.

Правильный результат для операций с плавающей запятой получается, если $0 \leq x_a < 8 \cdot /x_s/$.

Если эта команда используется для арифметических операций "с фиксированной запятой", т.е., когда программист совсем не принимает во внимание действий с порядками, правильный результат может быть получен только в том случае, если $0 \leq x_a \leq /x_s/ < I,0$, но не $x_s = -I,0$.

Команда 376

x_s проверяется на нуль или на недонормализованность и, если это так, происходит прерывание. Если прерывания не происходит, то деление продолжается, как в команде 375.

Команда 377

Та же, что и команда 376, но начальное содержимое I очищается.

3.2. ПЕРЕПОЛНЕНИЕ И ВЫХОД ИЗ РАЗРЯДНОЙ СЕТКИ В СТОРОНУ МЛАДШИХ РАЗРЯДОВ

(а) Переполнение порядка числа

Если в результате арифметической операции порядок числа превышает +I27, то устанавливается триггер пополнения порядка числа (разряд 29, строка I ЗУV Центральной вычислительной машины) и производится автоматический ввод в контролирующую программу путем обычного процесса прерывания. В случае умножения, где промежуточный результат перед нормализацией не является нормализованным и порядок числа превышает +I27, тогда, при условии, что окончательный результат находится в допустимых пределах, пополнения порядка числа не происходит. Например, если при умножении содержимое М и X_s есть I/4 и $y_a = y_s = 64$, промежуточный ответ будет (I/I6, I28), а окончательный нормализованный ответ - (I/2, I27). Это не вызовет пополнения порядка.

Предлагается, чтобы при обычных условиях содержимое некоторых определенных регистров выводилось в печатном виде, после чего работа программы заканчивалась. Однако представляется возможным, чтобы программа назначала выполнение альтернативной программы. Она будет составлять часть выполняемой программы. Ввод в автоматическую программу посмертного анализа или программирующую программу будет управляться или путем установления В-регистра, или, возможно, путем выполнения команды перехода, хранимой в отдельном регистре ЗУ.

(в) Переполнение накапливающего сумматора (A0)

Если обнаруживается пополнение накапливающего сумматора (т.е. сумматор сверхнормализуется) при завершении арифметической операции при использовании ненормализованных команд без округления, тогда триггер пополнения сумматора устанавливается в ЗУ V (разряд 27, строка 6 Центральной вычислительной машины). Это не вызывает автоматического прерывания. Команды экстракода предусматриваются так, чтобы этот регистр можно было проверить и очистить. Разряд пополнения может быть установлен только при помощи операций накапливающего сумматора, а затем при помощи программы постоянного ЗУ может быть установлен в первоначальное положение.

Следует заметить, что при этих командах, если пополнение накапливающего сумматора не происходит, денормализация не имеет места.

(с) Выход порядка числа из разрядной сетки

Если в ходе арифметической операции порядок числа меньше, чем -I28, содержимое накапливающего сумматора заменяется нулем с плавающей запятой, т.е. $0 \cdot 8^{-I28}$ ($x_a = 0, y_a = -I28$).

В случае умножения, где содержимое М и X_s равны -I,0, а сумма порядков числа равна -I29, окончательный результат будет $x_a = 0, y_a = -I28$, вместо $x_a = I/8, y_a = -I28$.

3.3. ОСНОВНЫЕ В-ОПЕРАЦИИ

Все арифметические операции осуществляются по модулю 2^{24} . Разряд В-переноса устанавливается или очищается при помощи команд, обозначенных Вs. Он устанавливается большинством арифметических команд на значение переноса, формируемое в старшем разряде сумматора В.

S - всегда 24-разрядный регистр, который может быть или старшей или младшей половиной регистра 48-разрядного слова; два самых младших разряда S не принимаются во внимание.

Некоторые из следующих кодов отмечаются как "незадействованные", а в том случае, если они указываются и выполняются, то содержимое любого из регистров В, регистров основного ЗУ или тестовых регистров В изменяться не будет. Эти коды выполняют, однако, некоторые операции в сумматоре В и могут, следовательно, устанавливать или очищать разряд переноса В. Коды, отмеченные "n-типом", могут быть использованы в качестве ложных программ. Коды, отмеченные "S-типом", требуют компоненту операции, определяемую в разрядах адреса, и могут таким образом вызвать прерывание "неэквивалентности" или "безусловное нарушение" при помощи компоненты операции. Прерывания можно избежать при помощи этих кодов путем определения адреса постоянного ЗУ (т.е. старшего восьмеричного разряда адреса -4) или адреса слова, находящегося в ЗУ на сердечниках (например, текущая команда).

- 0100 $ba' = s - ba$ Vc
Вычесть ba из s и поместить результат в Va . Содержимое S не меняется.
- 0101 $ba' = s$
Переписать 24 разряда из S в Va .
- 0102 $ba' = ba - s$ Vc
Вычесть s из ba и поместить результат в Va .
- 0103 $ba' = -s$
Поместить s с отрицательным знаком в Va .
- 0104 $ba' = ba + s$ Vc
Добавить s к ba и поместить результат в Va .
- 0105 $ba' = ba$ (с циклическим сдвигом вверх на 6 разрядов) + s
Сдвинуть содержимое Va вверх на 6 разрядов, передавая начальные старшие 6 разрядов в новые младшие 6 разрядов, затем прибавить s и поместить результат в Va .
- 0106 $ba' = ba \neq s$ очищает перенос V .
Определяет неэквивалентность ba и s , помещает результат в Va .
- 0107 $ba' = ba \& s$
Логически перемножить ba и s и поместить результат в Va .
- 0110 $s' = s - ba$ Vc
Вычесть ba из s и поместить результат в S . Содержимое Va не изменяется.
- 0111 $s' = -ba$
Поместить ba с противоположным знаком в S .
- 0112 $s' = ba - s$ Vc
Вычесть s из ba и поместить результат в S .
- 0113 $s' = ba$
Передать ba в S .
- 0114 $s' = ba + s$ Vc
Прибавить ba к s и поместить результат в S .
- 0115 То же, что и 0114
- 0116 $s' = ba \neq s$ очищает перенос V .
Определяет неэквивалентность ba и s и помещает результат в S .
- 0117 $s' = ba' \& s$
Логически умножить ba и s и поместить результат в S .
- 0120 $ba' = n - ba$ Vc
Вычесть ba из целого числа n и поместить результат в Va .
- 0121 $ba' = n$
Поместить целое число n в Va .
- 0122 $ba' = ba - n$ Vc
Вычесть целое число n из ba и поместить результат в Va .
- 0123 $ba' = -n$
Поместить целое число n с отрицательным знаком в Va .
- 0124 $ba' = ba + n$ Vc
Прибавить целое число n к ba и поместить результат в Va .

- 0125 $ba' = ba$ (с циклическим сдвигом вверх на 6 разрядов) + n
Сдвинуть содержимое Va вверх на 6 разрядов, переписывая начальные старшие 6 разрядов в новые младшие 6 разрядов, затем прибавить целое число n и поместить результат в Va .
- 0126 $ba' = ba \neq n$ очищает Vc
Определяет неэквивалентность ba и целого числа n и помещает результат в Va .
- 0127 $ba' = ba \& n$
Логически перемножить ba и целое число n и поместить результат в Va .
- 0130 недействительный код типа $n(n - ba)$ Vc
- 0131 недействительный код типа $n(-ba)$
- 0132 недействительный код типа $n(ba - n)$ Vc
- 0133 недействительный код типа $n(ba)$
- 0134 недействительный код типа $n(ba + n)$ Vc
- 0135 недействительный код типа $n(ba + n)$
- 0136 недействительный код типа $n(ba \neq n)$ очищает Vc
- 0137 недействительный код типа $n(ba \& n)$
- 0140 то же, что и 0100
- 0141 то же, что и 0101
- 0142 то же, что и 0102
- 0143 $ba' = ba$ (с циклическим сдвигом вниз на 1 разряд) - s
Сдвинуть содержимое Va вниз на 1 разряд, переписывая начальный младший разряд в старший разряд, вычесть s' и поместить результат в Va .
- 0144 то же, что и 0104
- 0145 то же, что и 0105
- 0146 $ba' = ba \vee s$, очищает Vc
Осуществить логическое "или" над ba и s и поместить результат в Va .
- 0147 то же, что и 0146, но Vc не изменяется
- 0150 $bt' = s - ba$ Vc
Установить bt в соответствие с результатом вычитания Va из S .
Содержимое ba и S не изменяются.
- 0151 недействительный код типа $s(s - ba)$
- 0152 $bt' = ba - s$ Vc
Установить bt в соответствие с результатом вычитания s из ba .
- 0153 недействительный код типа $s(ba - s)$
- 0154 код типа s , ($ba + s$) Vc не приспан
- 0155 недействительный код типа s ($ba + s$)
- 0156 недействительный код типа s ($ba \vee s$) очищает Vc
- 0157 недействительный код типа s ($ba \vee s$)
- 0160 то же, что и 0120
- 0161 то же, что и 0121
- 0162 то же, что и 0122
- 0163 $ba' = ba$ (с циклическим сдвигом вниз на один разряд) - n
Сдвинуть содержимое Va вниз на один разряд, переписать начальный младший разряд в новый старший разряд, вычесть целое число n и поместить результат в Va .

0164 $ba' = ba + (bm \& n) Vc$
Логически перемножить bm и n , затем прибавить результат к ba , поместив этот окончательный результат в Va .

0165 $ba' = bm \& n Vc$
Логически перемножить bm и n , поместив результат в Va .

0166 $ba' = ba \vee n$ очищает Vc
Осуществить логическое "или" над ba и n , затем поместить результат в Va .

0167 То же, что и 0166, но Vc не изменяется.

0170 $bt' = n - ba Vc$
Установить Vt в соответствие с результатом вычитания ba из целого числа n .

0171 недействующий код типа $n(n - ba)$

0172 $bt' = ba - n Vc$
Установить Vt в соответствие с результатом вычитания целого числа n из ba .

0173 недействующий код типа $n (ba - n)$

0174 недействующий код типа $n (ba + n) Vc$

0175 недействующий код типа $n (ba + n)$

0176 недействующий код типа $n (ba \vee n)$ очищает Vc

0177 недействующий код типа $n (ba \vee n)$.

3.4. ОСНОВНЫЕ ТЕСТОВЫЕ ОПЕРАЦИИ

Поскольку 19-й разряд функции не декодируется для тестовых операций, для каждой операции имеется два кода. В некоторых случаях имеются также коды-дубликаты.

Содержимое Va не изменяется этими командами, если проверка не удалась. Если Va - регистр управления, используемый в данное время, содержимое увеличивается на 1 обычным путем и затем, если проверка удалась, заменяется на n .

В командах 0200-0207 и 0220-0223:

(а) bm увеличивается на 0,4 или 1,0, чтобы можно было выбрать следующее 24- или 48-разрядное слово. Это имеет место, только если проверка удалась,

(б) если $Va = Vm$, окончательное содержимое определенного регистра V есть n , если проверка успешна, т.е. изменение на 0,4 или 1,0 переписывается.

В командах 0224-0237 разряды Vm используются для модификации V как обычно.

0200 Если $bm \neq 0$, то $ba' = n$ и $bm' = bm + 0,4$

0240 Если содержимое Vm не нуль, поместить целое число n в Va и прибавить 0,4 к содержимому Vm . В противном случае, оставить Vm и Va без изменения.

0201 Если $bm \neq 0$, то $ba' = n$ и $bm' = bm + 1,0$

0241 Если bm - не нуль, поместить целое число n в Va и прибавить 1,0 к bm

0202 Если $bm \neq 0$, то $ba' = n$ и $bm' = bm - 0,4$

0242 Если bm не равно нулю, поместить целое число n в Va и вычесть 0,4 из bm

0203 Если $bm \neq 0$, то $ba' = n$ и $bm' = bm - 1,0$

0243 Если bm не равно нулю, поместить целое число n в Va и вычесть 1,0 из bm

0204 } То же, что и 0200

0244 } То же, что и 0201

0205 } То же, что и 0202

0245 } То же, что и 0202

0206 } То же, что и 0203

0246 } То же, что и 0203

0207 } То же, что и 0203

0247 } То же, что и 0203

0210 } Если bm нечетное, то $ba' = n$

0250 } Если младший двоичный разряд bm - единица, поместить целое число n в Va

0211 } Если bm четное, то $ba' = n$

0251 } Если младший разряд bm - нуль, поместить целое число n в Va

0212 } То же, что и 0210

0252 } То же, что и 0210

0213 } То же, что и 0211

0253 } То же, что и 0211

0214 } Если $bm = 0$, то $ba' = n$

0254 } Если $bm = 0$, поместить целое число n в Va

0215 } Если $bm \neq 0$, то $ba' = n$

0255 } Если bm - не нуль, поместить целое число n в Va

0216 } Если $bm \geq 0$, то $ba' = n$

0256 } Если старший разряд bm - нуль, поместить целое число n в Va

0217 } Если $bm < 0$, то $ba' = n$

0257 } Если старший разряд $bm - 1$, поместить целое число n в Va

0220 } Если $bt \neq 0$, то $ba' = n$ и $bm' = bm + 0,4$

0260 } Если Vt - не нуль, поместить целое число n в Va и прибавить 0,4 к bm
В противном случае, оставить Va , Vm без изменения

0221 } Если $bt \neq 0$, то $ba' = n$ и $bm' = bm + 1,0$

0261 } Если Vt - не нуль, поместить целое число n в Va и прибавить 1,0 к bm

0222 } Если $bt \neq 0$, то $ba' = n$ и $bm' = bm - 0,4$

0262 } Если Vt - не нуль, поместить целое число n в Va и вычесть 0,4 из bm

0223 } Если $bt \neq 0$, то $ba' = n$ и $bm' = bm - 1,0$

0263 } Если Vt не нуль, поместить целое число n в Va и вычесть 1,0 из bm

0224 } Если $bt = 0$, то $ba' = n$

0264 } Если Vt - нуль, поместить целое число n в Va

0225 } Если $bt \neq 0$, то $ba' = n$

0265 } Если Vt - не нуль, поместить целое число n в Va

0226 } Если $bt \geq 0$, то $ba' = n$

0266 } Если Vt получает значение ≥ 0 , поместить целое число n в Va

0227 } Если $bt < 0$, то $ba' = n$

0267 } Если Vt получает значение < 0 , поместить целое число n в Va

0230 } То же, что и 0234

0270 } То же, что и 0235

0231 } То же, что и 0235

0271 } То же, что и 0235

- 0232} То же, что и 0236
 0272}
 0233} То же, что и 0237
 0273}
 0234} Если $xa=0$, то $ba'=n$
 0274} Если xa - нуль, поместить целое число n в Ba .
 Знаковый разряд L не проверяется
 0235} Если $xa \neq 0$, то $ba'=n$
 0275} Если xa не нуль, поместить целое число n в Ba .
 Знаковый разряд L не проверяется
 0236} Если $xa \geq 0$, то $ba'=n$
 0276} Если xa положительно или 0, поместить целое число n в Ba
 0237} Если $xa < 0$, то $ba'=n$
 0277} Если xa отрицательно, поместить целое число n в Ba .

3.5. ОСНОВНЫЕ ОПЕРАЦИИ НАКАПЛИВАЮЩЕГО СУММАТОРА

Все операции сумматора выполняются с числами с плавающей запятой. Однако некоторые команды могут быть использованы для операций с фиксированной запятой при условии, что в случае необходимости порядки чисел, над которыми производится операция, одинаковы (они, как правило, будут равны нулю).

Команды, включая арифметические операции, которые можно использовать для операций с фиксированной запятой, - это команды, устанавливающие, где это нужно, регистр переполнения накапливающего сумматора. Некоторые команды отмечаются как устанавливающие, где нужно, регистр переполнения порядка числа или регистр переполнения накапливающего сумматора; регистры будут установлены, даже если никаких случайностей не возникнет. Они отмечаются потому, что накапливающий сумматор проверяется на эти условия переполнения.

Для некоторых из следующих кодов компонента операции ЗУ не требуется (340, 341, 350, 351, 360, 361, 364-366, 370, 371). Обращение, однако, производится к регистру ЗУ, определяемому разрядами S в команде и, следовательно, может произойти прерывание "неэквивалентности" или "безусловного нарушения" при помощи компонента операции. Такого прерывания не происходит, если определяется адрес постоянного ЗУ (т.е. адрес, где старший восьмеричный разряд равен 4) или адрес слова, которое, как известно, находится в ЗУ на сердечниках (например, данная команда).

Детали методов выполнения некоторых операций накапливающего сумматора даны в разделе 3.1.

6300 EO $a' = am + s$ (Q)

Очистить L , включая знаковый разряд, и прибавить число с плавающей запятой в S к am . Результат нормализуется как число двойной длины.

Устанавливается регистр переполнения порядка числа и происходит прерывание в случае переполнения порядка.

0301 EO $a' = am - s$ (Q)

Если $ya \geq ys$, числу в S приписывается отрицательный знак и команда выполняется как 0300. Если $ya < ys$, число в S вычитается из числа, взятого из AM , и результат нормализуется как число двойной длины.

0302 EO $a' = -am + s$ (Q)

То же, что и 0301, но S и AM меняются ролями.

ПРИМЕЧАНИЕ. В командах 0300-0302 никакие совместные арифметические операции над обоими числами не будут выполняться, если порядки чисел отличаются на 32 и более; am устанавливается на число с большим порядком и этому числу приписывается отрицательный знак, если это указывается командой.

0303 То же, что и 0302

0304 EO $am' = s, l' = 0$ (Q)

Очистить L , включая разряд знака, затем передать содержимое ячейки памяти S в AM и нормализовать как число двойной длины. Никакой случайности не возникает, когда регистр переполнения порядка числа устанавливается этой командой.

0305

Очистить L , включая знаковый разряд, затем передать содержимое ячейки памяти S с противоположным знаком в AM и нормализовать как число двойной длины. Переполнение порядка числа происходит, если $ys = I27, xs = -I,0$.

0306 EO $am' = s, l' = 0$ (Q)

Подобно 0304, за исключением того, что используются схемы с "обратным минусом" в сумматоре.

Никакой случайности не возникает, когда происходит переполнение порядка.

0307 То же, что и 0306

0310 EO $a' = am + s$ (неочищенное L) (Q)

Как для 0300, кроме того, что L первоначально не очищается

0311 EO $a' = am - s$ (неочищенное L) (Q)

Как для 0301, кроме того, что L первоначально не очищается

ПРИМЕЧАНИЕ. Команды 0310 и 0311 имеет смысл применять для операций с удвоенным количеством разрядов только в том случае, если $ys \geq ya$.

0312 То же, что и 0302

0313 То же, что и 0302

0314 AO $am' = s$ (неочищенное L)

Передать число с плавающей запятой в ячейку памяти S в порядок числа и в старшую половину A . Младшая половина A остается без изменения.

Переполнение накапливающего сумматора не может возникнуть при этой операции.

0315 AO $am' = -s$ (неочищенное L)

Передать содержимое S с отрицательным знаком в AM . Регистр переполнения сумматора устанавливается, если содержимое $Xs = -I,0$. Начальное содержимое L не изменяется.

0316 То же, что и 0306

0317 То же, что и 0306

0320 EO $am' = am + s$ (QR)

Очистить L, включая знаковый разряд и прибавить число с плавающей запятой в S к числу с плавающей запятой в AM; результат нормализуется как число двойной длины и округляется путем введения единицы в младший разряд AM, кроме того случая, когда $L = 0$.

Если порядок ya переполняется, то устанавливается регистр переполнения порядка и происходит прерывание.

0321 EO $am' = am - s$ (QR)

Если $ya > ys$, числу в S приписывается отрицательный знак и команда затем выполняется как 0320.

Если $ya < ys$, число, находящееся в S, вычитается из числа, взятого из AM, нормализация и округление происходят как при 0320.

0322 EO $am' = -am + s$ (QR)

То же, что и 0321, но S и AM меняются ролями.

ПРИМЕЧАНИЕ. В 0320, 0321 и 0322, если порядки различаются на 16 или более, никакие арифметические операции над двумя числами сразу не производятся. Содержимое AM заменяется большим из 2 чисел, ставится знак -, если это указано в команде, и нормализуется. Округление может иметь место в любом случае, когда это требуется, например, если $ya - ys > 16$ и am содержит в M сверхнормализованное число в первоначальном виде с самым младшим восьмеричным разрядом, не равным нулю.

0323 То же, что и 0322

0324 То же, что и 0304

0325 То же, что и 0305

0326 То же, что и 0306

0327 То же, что и 0306

0330 AO $a' = am + s$

Очистить L, включая знаковый разряд и прибавить число с плавающей запятой в S к числу с плавающей запятой в AM. Если результат сверхнормализован, устанавливается регистр переполнения накапливающего сумматора.

0331 AO $a' = am - s$

Очистить L, включая знаковый разряд, после чего или сделать отрицательным число, взятое из S, и прибавить его к числу, взятому из AM, или вычесть число, взятое из S из числа в AM, в зависимости от того, меньше S или больше, или равно ya .

0332 AO $a' = -am + s$

То же, что и 0331, но S и AM меняются ролями.

ПРИМЕЧАНИЕ. В командах 0330-0332 никакие арифметические операции над обоими числами не выполняются, если порядки чисел различаются на 32 или больше, но am заменяется на число с более высоким порядком, и это число делается отрицательным, если это определяется командой.

0333 То же, что и 0332

0334 AO $am' = s$; $l' = 0$

Очистить L, включая знаковый разряд, и передать число с плавающей запятой в ячейку памяти

ПРИМЕЧАНИЕ. Эта команда отменяет AO, хотя она не

может в действительности установить регистр переполнения накапливающего сумматора.

0335 AO $am' = -s$, $l' = 0$.

Очистить L, включая знаковый разряд, затем поместить число в S, со знаком в старшую половину A. Порядок числа в S передается в Ya. Регистр переполнения накапливающего сумматора устанавливается, если $xs = -1, 0$.

0336 То же, что и 0306

0337 То же, что и 0306

0340 EO $a' = a$ (Q)

Нормализовать число двойной длины в A. Переполнение порядка числа может иметь место только если число в A было первоначально нормализовано и $ya = I27$.

0341 EO $a' = a$

Установить регистр переполнения порядка и сделать прерывание, если содержимое Ya больше, чем +I27.

0342 EO $am' = am.s$ (Q)

Образовать произведение двойной длины чисел в AM и S и нормализовать.

Если Ya переполняется, устанавливается регистр переполнения порядка и происходит прерывание.

0343 EO $a' = -am.s$ (Q)

То же, что 0342, но образуется отрицательное произведение;

0344 $l' = xs$, $ya' = ya$

Передать младшие 39 разрядов числа в S младшую половину A и знаковый разряд S в знаковый разряд L. Порядок числа A не изменится.

0345 $l' = xs$; $m' = \text{знак } xs$; $ya' = ya$

Передать младшие 39 разрядов числа, находящегося в S, в L, и передать знак этого числа в 40-й разряд M и в знаковый разряд. Порядок числа A не изменяется.

0346 $s' = am$; $a' = 0$

Передать am в S и очистить A до нуля с плавающей запятой, т.е. $ya = -I28$, M и L (включая знаковый разряд L) = 0.

0347 $s' = al$; $l' = 0$

Передать порядок числа A, знаковый разряд L и младшие 39 разрядов A в S; очистить L, включая знаковый разряд L.

0350 То же, что и 0340

0351 То же, что и 0341

0352 EO, AO $a' = am.s$; знак $l' = \text{знак } m'$

Образовать произведение с двойной длиной чисел в AM и S. Знаковый разряд L тот же самый, что и окончательный знак M. Переполнение накапливающего сумматора происходит, если начальное содержимое M и S равно -1, 0.

0353 EO, AO $a' = -am.s$; знак $l' = \text{знак } m'$
То же, что и 0352, но образуется отрицательное произведение. Результат проверяется на переполнение накапливающего сумматора, хотя оно не может произойти до тех пор, пока начальное содержимое A не будет сверхнормализовано.

0354 AO $am' = a(R+)$

Прибавить 1 к младшему разряду M, если старший разряд L равен единице. Содержимое L остается без изменения.

0355 EO $s' = am$, $am' = 1(Q)$

Передать порядок числа и старшую половину A в S, передать знак L в M и нормализовать A. Знак L не очищается. Производится проверка на переполнение, хотя оно не может произойти.

0356 $s' = am$

Передать порядок и старшую половину A в S.

0357 $s' = al$

Передать порядок A, знаковый разряд L и младшую половину A в S.

0360 EO $am' = a(QR)$

Нормализовать накапливающий сумматор двойной длины и округлить AM путем введения единицы в его самый младший разряд, за исключением того случая, когда окончательное содержимое L - нуль. Переполнение порядка может произойти только в том случае, если число в A было первоначально сверхнормализовано $ya = I27$.

0361 EO $am' = a(RO)$

Округлить AM путем введения единицы в самый младший разряд, кроме того случая, когда $l = 0$. Регистр переполнения порядка устанавливается и происходит прерывание, если содержимое Ya больше, чем $+I27$.

0362 EO $am' = am.s(QR)$

Образовать произведение с удвоенным количеством чисел в AM и S; нормализовать и округлить этот результат. Знак у L не изменяется.

Если Ya переполняется, устанавливается регистр переполнения порядка и происходит прерывание.

0363 EO $am' = -am.s(QR)$

То же, что и 0362, но образуется отрицательное произведение.

0364 AO $xa' = 8.xa$, $ya' = ya$

Сдвинуть содержимое накапливающего сумматора вниз на один восьмеричный разряд. Сдвиг снабжается знаком из старшего дополнительного разряда. Порядок числа не изменяется и переполнения не происходит.

0366 EO $am' = |am|(Q)$

Очистить L, взять дополнение к содержимому AM, если оно отрицательно, и нормализовать. Переполнение порядка числа устанавливается и происходит прерывание, если это нужно.

0367 EO $am' = |s|$

Очистить L, передать модуль содержимого S в AM и нормализовать накапливающий сумматор двойной длины. Определяется переполнение порядка числа и прерывание происходит, если нужно.

0370 То же, что и 0340

0371 То же, что и 0341

0372 EO, AO $a' = am.s$; знак $L' = 0$

Образовать произведение чисел с удвоенным количеством разрядов в AM и S. Очищается знаковый разряд L. Если Ya переполняется, установить регистр переполнения порядка числа и осуществить прерывание. Если сумматор

становится сверхнормализованным (т.е. в результате умножения 2-х чисел, оба аргумента которых $-1, 0$) устанавливается регистр переполнения накапливающего сумматора.

0373 EO AO $a' = -am.s$; знак $L' = 0$

То же, что и 0372, но образуется отрицательное произведение. Накапливающий сумматор проверяется на переполнение, хотя оно не может произойти до тех пор, пока начальное содержимое A не будет сверхнормализовано.

0374 EO, DO $am' = am + s(QR)$

Очистить L и разделить am на s ; результат округляется путем введения единицы в самый младший разряд частного перед нормализацией, если имеется остаток. Происходит прерывание, если s недонормализовано или равно нулю (разряд DO в строке I ЗУ V Центральной вычислительной машины) или если происходит переполнение порядка числа.

0375 EO $al' = a + |s|$, где a должно быть положительно, ya - порядок частного.

am' - положительный остаток от $a + |s|$, где его порядок берется:

(а) $ya - I2$, если $|xs| \leq xa$

(б) $ya - I3$, если $|xs| > xa$

Прерывание происходит, если имеет место переполнение порядка числа. Прерывание не происходит, если s недонормализовано или равно нулю. Правильный результат для операций с плавающей запятой получается, если

$$0 < xa < 8 \cdot |xs|$$

Если эта команда используется для арифметических операций с "фиксированной запятой", т.е. когда программист не принимает во внимание арифметические операции над порядком числа, правильный ответ получается, если

0376 EO, DO $al' = a + |s|$, где a должно быть

положительным

ya' = порядок частного

m = остаток $a + |s|$, где его порядок берется

(а) $ya - I2$, если $|xs| \leq xa$

или

(б) $ya - I3$, если $|xs| > xa$

Прерывание происходит, если s недонормализовано или равно нулю или если происходит переполнение порядка числа.

0377 EO, DO $al' = |am| + |s|$, ya' - порядок числа частного

m' = остаток $|am| + |s|$, где его порядок берется:

ся:

(а) $ya - I2$, если $|xs| \leq |xa|$

(б) $ya - I3$, если $|xs| > |xa|$

Прерывание происходит, если s недонормализовано или равно нулю, или если происходит переполнение порядка числа.

3.6. ОРГАНИЗИРУЮЩАЯ ПРОГРАММА

3.6.1. КООРДИНИРОВАНИЕ ПОДПРОГРАММ

Структура организующей программы

Организирующая программа управляет всеми теми функциями системы, которые не предусмотрены в основной программе⁺ Центральной вычислительной машины или не присущи функциям внешнего оборудования. Следовательно, организующая программа используется довольно часто, в том числе, всякий раз, когда какая-либо часть системы требует ее внимания. Она приводится в действие несколькими различными способами. Во-первых, она может быть введена в действие в результате работы основной программы. Таким образом, выполняемая задача требует вмешательства организующей программы всякий раз, когда происходит, например, обращение к внешнему оборудованию, или к запоминающим устройствам на сердечниках и магнитных барабанах. Организующая программа вступает в работу также тогда, когда основная программа по ряду причин требует наблюдения за операциями, выполнением действий с порядком числа или переполнением при делении, или переполнением запоминающего устройства, или за распределением времени. Во-вторых, организующая программа может быть использована различными устройствами машины, которые выполняли свои определенные задания и требуют дальнейшего внимания. Организующая программа, например, начинает работать от команд барабанов и магнитных лент, при которых происходит передача "блока" из 512 слов в запоминающее устройство на сердечниках или из него. Внешнее оборудование требует внимания всякий раз, когда происходит заполнение или опустошение буферного запоминающего устройства на один или ряд знаков с помощью этого оборудования. Наконец, повреждения Центральной вычислительной машины, запоминающего устройства и внешнего оборудования вызывают работу организующей программы. Центральная вычислительная машина, таким образом, распределяет свое время между работой организующей программы и основных программ, причем программирование машины "Атлас" таково, что исключено взаимное воздействие основных программ и всех частей организующей программы. Организующая программа имеет много ответвлений, которые обычно являются бездействующими, но

⁺) Программа решаемой задачи

могут быть приведены в действие по первому требованию. Последовательность, с которой ответвления программы приводятся в действие, является совершенно произвольной, поскольку это определяется работой основной программы и внешнего оборудования.

Подпрограммы прерывания

Наиболее часто и быстро используемые части организующей программы — подпрограммы прерывания. Например, когда внешнее оборудование требует внимания, то устанавливается триггер прерывания, который представляет собой разряд запоминающего устройства "v" в Центральной вычислительной машине, причем предусмотрен отдельный триггер прерывания для каждого случая, когда необходимо прерывание. Если триггер прерывания устанавливается и прерывания не запрещены, тогда до начала следующей команды в регистр В125 управления прерыванием записывается адрес 2048 из постоянного запоминающего устройства, и управление переключается на прерывание. Дальнейшие прерывания запрещаются до тех пор, пока управление не перейдет к главному управлению или управлению "экстракодами". При управлении прерыванием программа из постоянного запоминающего устройства, которая начинается с адреса 2048, определяет, какой триггер прерывания был установлен, и запускает соответствующую программу прерывания из постоянного запоминающего устройства. Если устанавливается более одного триггера и оспаривается очередность первого, то эта очередность устанавливается в зависимости от степени срочности требуемого действия. Используя специальную аппаратуру, относящуюся к одному из В-регистров, например В123, можно определить источник любого прерывания в результате выполнения от двух до шести команд.

Таким образом, вводимые программы прерывания имеют дело с непосредственной причиной данного прерывания. Например, когда буферное устройство на один знак, связанное с устройством считывания с бумажной ленты, заполнится, устанавливается соответствующий триггер прерывания и вводится "программа прерывания устройства считывания с бумажной ленты". Эта программа передает знак в нужную ячейку запоминающего устройства после контроля по четности, где это необходимо. Тем временем устройство, считывающее с бумажной ленты, приступает к считыванию следующего знака в буферное устройство. Отдельные программы прерывания в постоянном запоминающем устройстве управляют каждым типом внешнего оборудования, магнитными лентами и барабанами. Эта техника прерывания также используется в некоторых необычных ситуациях, которые случаются, когда сама Центральная вычислительная машина не может работать одинаково над задачей, в условиях, когда, например, имеется переполнение или же когда нет в данный момент нужного блока в запоминающем устрой-

стве на сердечниках. Поэтому для работы в подобных случаях имеются триггеры прерывания и программы прерывания. Другие программы обеспечивают прерывания, связанные с обнаружением ошибок вычислительной машины.

Во время работы программы прерывания другие прерывания запрещаются, и триггеры прерывания в памяти "v" остаются в установленном состоянии. При возобновлении работы главного управления или управления "экстракодами", прерывания вновь разрешаются. Если тем временем устанавливается один или несколько триггеров прерывания, соответствующие программы прерывания выполняются в соответствии со степенью их значимости. Чтобы не мешать выполнению основных программ или организующих программ, программы прерывания используют только ограниченные устройства Центральной вычислительной машины, а именно: регистр управления прерыванием, строки "B" I23-я и от III по II8 включительно, частные регистры во вспомогательном запоминающем устройстве и запоминающем устройстве "v", "запретные" страницы запоминающего устройства на сердечниках (см.раздел 3). За исключением строк "B", ни одной основной программе не разрешается использовать эти регистры. На строки "B" не накладывается никакого запрета, кроме того, что в программах прерывания не делается никаких предположений относительно первоначального содержимого строк "B", и, следовательно, в самом худшем случае, ошибочное использование основной программой строк "B", относящихся к прерыванию, может иметь результатом только ошибочное функционирование этой отдельной программы. Переключение управления на программу прерывания и наоборот происходит быстро, так как никакого сохранения или восстановления содержимого работающих регистров не требуется.

Программы прерывания составлены таким образом, что выполняются с минимальной задержкой и за самый короткий промежуток времени. Например, передача информации знак за знаком во внешнее оборудование и из него происходит с большой скоростью и необходимо, чтобы передача выполнялась при минимально возможном использовании Центральной вычислительной машины и в пределах отрезка времени, допускаемого внешним оборудованием для заполнения или считывания с буферного запоминающего устройства. Так как одновременно могут быть установлены несколько триггеров прерывания, но не могут начать действовать до тех пор, пока какая-либо программа прерывания еще выполняется, необходимо, чтобы каждая программа прерывания выполнялась в течение малого промежутка времени. Большинство запросов программ прерывания состоит только из нескольких команд, таких как передача разряда, подсчет количества шагов и т.д., и в заключение, программа прерывания возвращается к прежнему управлению или главному или экстракодам. Однако в некоторых случаях требуются более длинные последовательности команд, например, по окончании ввода бумажной перфоленты или комплекта карт должны быть введены программы для того, чтобы оперировать с разрядами, собранными в запоминающем устройстве, записы-

вать их на магнитную ленту соответствующим образом, декодировать и составлять заголовки и т.д. В таких случаях программа прерывания вводит программу, которая подчиняется управлению экстракодами, известному как программа управления экстракодами.

Подпрограммы экстракода организующей программы

Подпрограммы экстракода организующей программы образуют основные "ответвления" организующей программы. Они приводятся в действие либо программами прерывания, либо командами экстракода, встречающимися в основной программе. Они не перекрываются с основными программами, поскольку основное запоминающее устройство используется в качестве рабочего пространства вместе с областями запоминающих устройств на барабанах и магнитных сердечниках, которые блокируются обычным путем, пока выполняется основная программа (см.раздел 3). Они работают от управления экстракодами, причем регистр управления экстракодами любой текущей основной программы сохраняется и, следовательно, восстанавливается. Подобно программам прерывания они используют собственные строки "B", в таком случае, строки "B" от I00 до II0 включительно. Если требуются какие-либо другие рабочие регистры, сами организующие программы сохраняют и, следовательно, восстанавливают содержимое таких регистров. Подпрограммы экстракода организующей программы, таким образом, исключают перекрытие с дополнительной программой.

Эти ответвления организующей программы могут использоваться с любыми интервалами. Кроме того, они могут прерываться подпрограммами прерывания, которые, в свою очередь, могут вызывать работу других подпрограмм экстракода организующей программы. Таким образом, можно привести в действие несколько подпрограмм экстракода организующей программы одновременно, так же как можно установить несколько триггеров прерывания в одно и то же время. Несмотря на то, что можно использовать несколько подпрограмм экстракода организующей программы, ясно, что выполнять в какой-то отрезок времени можно только одну программу, а остальные либо останавливаются, либо ожидают выполнения. Это осуществляется частью устройства управления, называемой координирующей подпрограммой, которая находится в постоянном запоминающем устройстве. Работа на подпрограмме экстракода организующей программы осуществляется через координирующую подпрограмму, которая организует это таким образом, что любая подпрограмма экстракода организующей программы, уже приведенная в действие, не прерывается другими подпрограммами экстракода организующей программы. По мере их использования они записываются в "списки" во вспомогательном запоминающем устройстве, и запись извлека-

ется из одного из этих "списков" всякий раз, когда подпрограмма экстракода организующей программы заканчивается или останавливается сама. Уже начавшись, подпрограмма экстракода всегда имеет возможность продолжаться, если это возможно, причем более важная по значению программа экстракода не прерывает менее важной по значению подпрограммы экстракода, а вводится только в конце или при останове текущей подпрограммы экстракода. Координатор играет роль программы, равноценной "триггеру, запрещающему прерывание", причем списки приведенных в действие подпрограмм экстракода организующей программы являются равноценными установке нескольких триггеров прерывания. Два основных различия состоят в том, что никакого лимита времени для подпрограмм экстракода организующей программы не устанавливается и что программа может сама останавливаться по различным причинам; в этом она отличается от программы прерывания, которые соблюдают лимит времени и никогда не останавливаются.

Для того, чтобы работа каждого ответвления вычислительной системы поддерживалась по возможности на самом высоком уровне, подпрограммы экстракода организующей программы, ожидающие выполнения, записываются в четырех различных "списках". В пределах каждого списка программы располагаются в том порядке, в каком они должны выполняться, но сами списки располагаются по очередности действий, так что более важный по значению список выполняется прежде, чем производится обращение к следующему списку. Более важный по значению список содержит программы, введенные с барабана, и также программы, введенные в результате сбоя вычислительной машины, например, в результате контроля по четности запоминающего устройства на сердечниках. Второй список содержит программы, являющиеся результатом прерываний работы магнитной ленты, и третий содержит программы, являющиеся результатом прерываний работы внешнего оборудования. Самый менее важный по значению список содержит один ввод для каждой основной программы, которая в данный момент выполняется. В этот список также записывается и ввод к подпрограмме экстракода организующей программы через команду экстракода в основную программу. По окончании подпрограммы экстракода координирующая программа выбирает для исполнения первую приведенную в действие подпрограмму экстракода в самом старшем по значению списке.

Необязательно, чтобы Центральная вычислительная машина была полностью занята во время выполнения подпрограмм экстракода организующей программы. Подпрограмма может, например, требовать передачи "блока" информации с барабана в запоминающее устройство на сердечниках, в таком случае она останавливается, пока не окончится передача с барабана. Кроме того, очередь запросов на обмен с магнитными барабанами (см. раздел 3), содержащаяся во вспомогательном запоминающем устройстве, может быть полной, в этом случае подпрограмма экстракода, делающая запрос, должна быть остановлена. Когда подпрог-

рамма экстракода останавливается по этой или подобной причине, она возвращается к соответствующему списку, а следующая подпрограмма вводится координирующей программой. Прежде чем подпрограмма экстракода остановится, определяется момент нового запуска. Остановленная подпрограмма запускается снова, когда причина останова будет устранена, например, с помощью подпрограммы экстракода, которая управляет обращением к барабанам и извлечением записей очередности с барабана. Следовательно, списки подпрограмм экстракода могут содержать в любой момент программы, ожидающие выполнения и остановленные программы; программы прерывания записаны таким образом, что количество подпрограмм экстракода, задействованных в любой момент времени, ограничивается до одной на основную программу и одной или двумя на один триггер прерывания в зависимости от особенностей каждой программы прерывания. Когда в конце концов устанавливается, что подпрограмма экстракода отличается от остановленных программ, она "вычеркивается" из списков подпрограмм экстракода и снова становится пассивной.

Несмотря на то, что подпрограммы экстракода организующей программы часто составляются как подпрограммы для управления внешним оборудованием, магнитными лентами и барабанами, не следует предполагать, что это является единственной функцией данных подпрограмм. Входы к подпрограммам экстракода из программ прерывания или команд экстракода в основной программе дают начало программам, которые управляют всей работой вычислительной системы, включая передачу информации между запоминающим устройством и внешним оборудованием, связь с операторами и инженерами, начало, окончание и, где необходимо, контроль основной программы Центральной вычислительной машины и сбоя в работе внешнего оборудования, выполнения тестовых программ и накопление информации о работе. Каждая разновидность работы организующей программы составляется из ряда подпрограмм экстракода, причем каждая такая подпрограмма запускается основной программой или программой прерывания и заканчивается обычно обращением к внешнему оборудованию или магнитной ленте или же изменением списка подпрограмм экстракода или списка основных программ. Наиболее часто используемые программы хранятся в постоянном запоминающем устройстве; программы, которые используются реже, хранятся на магнитном барабане и передаются по требованию в запоминающее устройство на сердечниках. На организующие подпрограммы в запоминающих устройствах на сердечниках и на барабанах не оказывают влияния основные программы благодаря схемам блокировки и подпрограммам в постоянном запоминающем устройстве.

Основные программы

Функция всей организующей программы состоит, конечно, в том, чтобы организовывать прохождение задач через вы-

числительную машину по возможности с минимальной задержкой. Основные программы вводятся подпрограммами экстракода организующей программы, которые включают их в список основных программ; впоследствии они эффективно вводятся координирующей программой как ответвления меньшей значимости, чем любая подпрограмма экстракода. Хотя основные программы логически являются подпрограммами организующей программы, они могут функционировать долгое время самостоятельно без обращения к организующей программе. Исходя из этого, организующую программу можно рассматривать как обычную пассивную программу, которая работает и использует Центральную вычислительную машину только малую часть имеющегося в распоряжении времени.

Чтобы дать возможность основным программам функционировать с минимальным вмешательством организующей программы, им не разрешается использовать непосредственно управление экстракодами и прерывания; это позволяет оградить главные программы от воздействия организующих программ. Основные программы используют главный регистр управления "B" I27, и поэтому им запрещается обращаться к запоминающему устройству "V" и основному запоминающему устройству. Обращение к любому из этих запоминающих устройств вызывает установку триггеров прерывания и, следовательно, переход на организующую программу.

Выборка из частных запоминающих устройств получается только косвенным путем при использовании функций экстракода, которые переключают данную программу на управление экстракодами и вводят одну из максимально возможных 512 программ в постоянном запоминающем устройстве. Эти программы экстракода формируют простые изменения основного кода операций, а также обеспечивают определенный ввод в организующие подпрограммы, чтобы управлять передачей информации в запоминающее устройство на сердечниках и из него и проводить необходимую организацию работ. Такие специфические вводы в организующую программу полностью предохранят основные программы от влияния других программ. Блоки запоминающего устройства на сердечниках и барабанах защищаются при помощи аппаратуры и организующими подпрограммами в постоянном запоминающем устройстве, как описано в разделе 3.

Основная программа останавливается (при помощи подпрограмм экстракода организующей программы) всякий раз, когда требуется обращение к блоку информации, которого в данный момент в запоминающем устройстве на сердечниках нет. Этот блок может находиться на барабане и в этом случае вводится программа обращения к барабану или же может понадобиться обращение к магнитной ленте. В обоих случаях программа останавливается и не выполняется до тех пор, пока блок информации не появится в запоминающем устройстве на сердечниках. В случае информации, требующей обращения к внешним устройствам, такой, как входные данные или выходные результаты, организующая программа создает буфер для информации в запомина-

ющем устройстве на магнитных сердечниках и барабанах, и "непосредственное" управление внешним оборудованием с помощью основной программы не допускается. В этом случае можно избежать неиспользования больших "секций" информации в запоминающем устройстве в то время, как программа ожидает обращения к внешнему оборудованию. Однако программа может непосредственно управлять обращением к барабанам и магнитным лентам, используя функции экстракода, которые вызывают обращение к соответствующим подпрограммам организующей программы. Последовательности команд отсылаются в основное запоминающее устройство с помощью этих подпрограмм, чтобы дать возможность основной программе продолжать и успешно осуществлять, по возможности, наиболее полное перекрытие при обращении к барабану и ленте и выполнением основной программы.

В то время, как одна программа останавливается, ожидая например, окончания обращения к магнитной ленте, координирующая программа переключает управление на следующую программу в списке основных программ, которая готова для продолжения работы. Для полной защиты от неправильного срабатывания нужно сохранять и восстанавливать содержимое рабочих регистров, общих для всех программ, таких как строки "B", накапливающий сумматор и регистры управления, и защитить блоки информации, используемые в запоминающем устройстве на сердечниках. Подпрограмма экстракода организующей программы для осуществления этого переключения от одной основной программы к другой занимает Центральную вычислительную машину на время, приблизительно равное $750 + 12p$ мксек, где "p" представляет собой количество "страниц", или блоков по 512 слов, в запоминающем устройстве на сердечниках. В вычислительной машине "Атлас" Манчестерского университета, которая имеет 32 "страницы" в запоминающем устройстве на сердечниках, время переключения с одной программы на другую и обратно составляет около 2,5 мсек. Это время можно сравнить примерно с 60 мксек, необходимыми для того, чтобы перейти на подпрограммы экстракода организующей программы и вернуться обратно, и даже меньшим временем для того, чтобы обратно перейти на программу прерывания и вернуться. Следовательно, ясно, что наиболее эффективным способом получения перекрытия между входом и выходом, обращениями к магнитной ленте и вычислением является сведение к минимуму количества изменений между основными программами и использование в полной мере быстро переключения на подпрограммы организующей программы и программы прерывания и обратно. Метод получения этого на практике описан в разделе 6.

Компиляция программ организующей программы является особым случаем выполнения основной программы, причем компилирующая программа содержит основную программу, которая обрабатывает программу языка источника как входные данные. Для компилируемых программ допускается особое оборудование, чтобы их распределение по объему запоминающего устройства можно было бы увеличить по мере необходимости, и чтобы позволить выход в систему управ-

ления до выполнения задачи или записи компилированной основной программы.

Обнаружение ошибок

В дополнение к программным обращениям к организующей программе, обращение может произойти и в случае некоторых обнаруживаемых ошибок, появившихся во время решения задачи. При исполнении программы может случиться ряд ошибок, которые обнаруживаются аппаратурой, контролем с помощью программы в экстракодах и организующей программой. При аппаратном контроле вызывается переход к организующей программе с помощью установки триггеров прерывания в случае переполнения накапливающего сумматора, использования неопределенной команды и обращения к основному запоминающему устройству или запоминающему устройству "v". Подпрограммы экстракода обнаруживают ошибки в области аргументов при извлечении квадратного корня, логарифма и команд \arcsin . В экстракодах, относящихся к внешнему оборудованию или магнитным лентам, контроль включается для того, чтобы логический номер оборудования был бы заранее определен. В экстракодах для перевода данных могут быть обнаружены ошибки в самих данных. Организующая программа обнаруживает ошибки при использовании запоминающего устройства. Все решаемые задачи должны сообщать организующей программе информацию о требуемом объеме памяти, количестве выходных результатов и предполагаемой длительности решения задачи. Эта информация подается до того, как программа компилируется или может быть выведена после компиляции. Организующая программа хранит запись об используемых блоках запоминающего устройства и может не допустить, чтобы программа превысила заданный лимит. Кроме того, триггер прерывания устанавливается с помощью синхронизатора с интервалами 0,1 сек, а другой триггер устанавливается всякий раз, когда выполняется 1024 команды, использующие главное управление или управление экстракодами. Эти триггеры вызывают обращения к организующей программе, которые дают возможность "контролировать" программу, чтобы убедиться, что предварительно введенный лимит времени не нарушен, и которые также способствуют введению программ для выполнения регулярных синхронизированных операций, таких как регистрация работы вычислительной машины и ввод тестовых программ.

Действие, принимаемое организующей программой, когда обнаруживается "ошибка" программы, зависит от условий, заранее установленных программой. Некоторые ошибки могут быть обнаружены отдельно, вызывая возвращение управления к предварительно заданному адресу. Может быть введена особая последовательность контролирующей программы, если требуется приспособить программу или компилирующую программу для получения диагностической печати. В случае отсутствия спецификации этих действий, информация печатается организующей программой, програм-

ма приостанавливается и обычно записывается на магнитную ленту, чтобы освободить запоминающее устройство для другой программы.

В следующих разделах подробно описывается работа некоторых подпрограмм организующей системы, а именно, подпрограмм для управления барабанами, магнитных лент и внешнего оборудования и подпрограмм, управляющих потоком информации в вычислительной машине.

3.6.2. СТРУКТУРА ЗАПОМИНАЮЩЕГО УСТРОЙСТВА

Косвенная выборка адреса и запоминающее устройство одного уровня

Запоминающее устройство на сердечниках вычислительной машины "Атлас" обеспечивается косвенной выборкой адресов, которая позволяет организующей программе перераспределять области запоминающего устройства и изменять их физические адреса, и которая также используется для выполнения автоматических обращений к барабану. С каждой "страницей", или блоком информации из 512 слов, запоминающего устройства на сердечниках связан "регистр адресов страниц", который содержит старшие разряды адресов блока информации, содержащейся на странице. Каждый раз, когда требуется выборка какого-либо слова информации из запоминающего устройства на сердечниках, страница, содержащая это слово, отыскивается аппаратным путем. При этом параллельно проверяется эквивалентность затребованного "адреса блока", или старших значащих разрядов адреса, и содержимое каждого из регистров адресов страниц. Если эквивалентность не имеет места, то это приводит к прерыванию решения задачи по "неэквивалентности". Регистры адресов страниц сами имеют адреса в запоминающем устройстве "v" и, следовательно, могут устанавливаться соответствующим образом с помощью организующей программы всякий раз, когда информация передается в запоминающее устройство на сердечниках или из него.

Одним из наиболее важных особенностей этой схемы является то, что она позволяет организующей программе осуществлять автоматическое обращение к барабанам. Адрес в команде относится к комбинированной памяти вычислительной машины на сердечниках и барабанах, а организующая программа записывает в основное запоминающее устройство адрес размещения каждого блока информации. Сохраняется только одна запись каждого блока, и адрес относится либо к странице запоминающего устройства на сердечниках, либо к сектору запоминающего устройства на барабанах. В любой момент времени только некоторые блоки, содержащие какую-либо особую программу, могут находиться в запоминающем устройстве на сердечниках, и если требуются только эти блоки, программа может выполнять-

ся на полной скорости. Когда требуется блок, которого нет в запоминающем устройстве на сердечниках, происходит прерывание по "неэквивалентности" и используется организующая программа для того, чтобы передать новый блок из сектора запоминающего устройства на барабане на страницу запоминающего устройства на сердечниках. Во время этой операции прерванная программа останавливается организующей программой.

"Справочник" блоков в основном запоминающем устройстве содержит один вход для каждого блока в комбинированной памяти на сердечниках и барабанах. Он делится на зоны для каждой основной программы, которая находится в запоминающем устройстве. "Справочник" отдельных программ определяет зону справочника блоков, занятую каждой программой. Размер этой зоны, или количество блоков, используемых одной программой, определяется до выполнения программы при описании работы (см. раздел 6). Ввод блока содержит номер блока "v" вместе с номером страницы или сектора, занимаемого блоком, и, если возможно, ставится в n-ую позицию в зоне, иначе зона поднимается рабочим материалом с конца. В этом случае блоки, используемые различными основными программами, всегда хранятся отдельно, не взирая на адреса, которые используются каждой программой. Программа обращается к комбинированному запоминающему устройству "одного уровня", и организующая программа при необходимости осуществляет обмен блоком информации между запоминающим устройством на сердечниках и барабанах. Физическое местоположение каждого блока информации не определяется программой, но контролируется организующей программой.

Имеются случаи, когда основная программа должна быть защищена от случайного обращения к странице запоминающего устройства на сердечниках, барабанах или лентах. Чтобы обеспечить полнейшую защиту таких страниц, каждый регистр адресов страниц имеет дополнительный разряд, известный как разряд блокировки. Он предотвращает выборку такой страницы вычислительной машиной, за исключением случая, когда работает управление прерыванием, и любое обращение к странице вызывает прерывание по "неэквивалентности". С помощью установки и сбрасывания разрядов блокировки организующая программа осуществляет полный контроль над использованием запоминающего устройства на сердечниках. Она может разрешить отдельным основным программам изменять содержимое запоминающего устройства на сердечниках, она может сохранить страницы для обменов с внешним оборудованием и может сама иногда использовать части запоминающего устройства на сердечниках для программ или рабочего пространства без риска вмешательства со стороны. Это делается с помощью такого расположения, что всякий раз, когда управление возвращается к основной программе, страницы, которые не находятся в ее распоряжении, блокируются.

Блок информации, образующий часть основной программы, может также быть заблокирован от использования этой программой, поскольку работа с информацией, контролируемой организующей программой, еще не закончена. Могут

потребоваться обращения к барабану, магнитной ленте или внешнему оборудованию, содержащим этот блок. Причина, по которой заблокирован такой блок, записывается в справочнике блоков, а если блок находится в запоминающем устройстве, на сердечниках, кроме того, устанавливается разряд блокировки. Если происходит обращение к такому блоку основной программы, то происходит прерывание работы по "неэквивалентности" и подпрограмма экстракода организующей программы останавливает программу. Эта подпрограмма экстракода снова приводится в действие программой координатора, когда блок становится "разблокированным", и дополнительная программа снова вводится, когда блок оказывается в запоминающем устройстве на сердечниках.

Программа обращения к барабану

Программа обращения к барабану представляет собой группу подпрограмм экстракода организующей программы, которые осуществляют обращения к барабанам и заполнение регистров адресов страниц и справочника блоков. Уже начавшись, передача полного блока информации на барабан или с барабана осуществляется схемным путем. Программа обращения к барабану начинает обращение и определяет нужный сектор барабана путем установки соответствующих разрядов в запоминающем устройстве "v". Она также определяет страницу запоминающего устройства на сердечниках, вводимую установкой особого "подставного" адреса блока, опознаваемого схемой управления барабана, в регистре адресов страниц; в тоже время эта страница блокируется, чтобы предупредить вмешательство основных программ, пока происходит передача информации.

По окончании обращения происходит прерывание работы, которое вводит программу обращения к барабану. Программа может также вводиться в подпрограммы прерывания по "неэквивалентности", которая определяет номер запрашиваемого блока, не найденного в регистрах адресов страниц. В конечном счете, использование программы обращения к барабану может определять другие части организующей программы, которые требуют обращения к барабану, и командами экстракода, которые обеспечивают метод, посредством которого основные программы могут, при желании, осуществлять некоторый контроль за продвижением блоков в запоминающее устройство на барабанах и из него. Очередность запросов для обращения к барабану запоминается в основном запоминающем устройстве, причем всего запоминается 64 запроса; когда программа обращения к барабану оканчивает передачу информации, начинается следующая по очереди передача.

Всякий раз, когда организующая программа предполагает осуществить новое обращение к барабану, возникает три возможных ситуации. Во-первых, список очередности запросов пуст и обращение к барабану может начинаться немедленно. Во-вторых, список очередности уже частично заполнен и запрос записывается в следующую позицию в списке очередности. В-третьих, список очередности за-

полнен. В этом случае, программа, делающая запрос, останавливается координирующей программой и возобновляется, когда в список очередности можно произвести новую запись. В первых двух случаях подпрограмма организующей программы заканчивается, когда запрос достигает списка очередности. При этом происходит прерывание работы по "неэквивалентности", подразумевающее требуемое обращение к барабану.

Запоминающее устройство на сердечниках организуется таким образом, чтобы всегда сохранять пустую страницу, не содержащую какой-либо полезной информации, и, когда требуется, на эту пустую страницу вводится блок информации с барабана. Пока осуществляется эта передача с барабана, производится подготовка к записи содержания другой страницы запоминающего устройства на сердечниках на барабан для того, чтобы сохранить пустую страницу. Выбор этой страницы представляет собой задачу "самообучающейся программы", которая содержит подробности работы в соответствии с полученной информацией. Эта самообучающаяся программа будет подробно описана в другом месте; она определяет страницу, которая не потребует в течение долгого времени. Программа организуется с обратной связью таким образом, что если она записывает блок, который почти сразу же снова запрашивается, она делает это только один раз. Номер выбранной страницы записывается в основное запоминающее устройство, и список последовательности обращений к барабану преобразуется для запроса, чтобы записать эту страницу на барабан. Эта подпрограмма организующей программы заканчивается и возвращает управление координирующей программе.

По окончании обращения к барабану снова вводится программа обращения к барабану. Она обращается к справочнику блоков и регистру адресов страниц, дает возможность работать основной программе и вводит следующий запрос барабана, при котором выбранная страница должна быть записана на барабан. После этого программа заканчивается, и снова вводится координирующая программа. В конечном счете, когда запись на барабан заканчивается, запускается организующая программа. Уточняется запись в справочнике блоков, делается запись о пустой странице и вводится следующий запрос для барабана.

Использование главного запоминающего устройства организующей программы

Некоторые подпрограммы организующей программы находятся в главном запоминающем устройстве, причем все эти программы используют рабочее пространство главного запоминающего устройства. Ввиду того, что организующая программа запускается без полной смены программ, особое внимание должно быть уделено тому, как хранить особым образом эти блоки информации запоминающего устройства и защищать их от воздействия со стороны других программ.

Активные блоки организующей программы главного запоминающего устройства записываются в области программы 0 в "справочнике блоков". Есть также блоки информации организующей программы, которые постоянно хранятся на барабане. Когда требуется один из этих постоянных блоков, он дублируется для того, чтобы создать активный блок организующей программы или, как в случае компилирующей программы, чтобы стать частью основной программы.

Из возможных 2048 номеров блоков информации 256 являются "резервными" номерами, которые используются исключительно организующей программой и не находятся в распоряжении основной программы. Основная программа ограничивается использованием остающихся "нерезервных" номеров блоков. Блоки с резервными номерами могут быть использованы организующей программой в любое время в запоминающем устройстве на сердечниках, и координирующая программа блокирует эти страницы запоминающего устройства на сердечниках перед передачей управления основной программе. Организующая программа также использует некоторые блоки, имеющие нерезервные номера для того, чтобы сохранить запись последовательности блоков информации, таких как потоки вводной и выводной информации. Когда в запоминающее устройство на сердечниках вызывается нерезервный блок для организующей программы, регистр адресов страниц не устанавливается, так как может быть блок основной программы, который имеет такой же номер блока, уже находящийся в запоминающем устройстве на сердечниках. Вместо этого, на регистре адресов страниц устанавливается фиксированный резервный номер блока, пока он используется, и очищается и блокируется прежде чем управление перейдет к другой программе.

Не все резервные номера блоков находятся в распоряжении организующей программы для обычного использования, так как некоторые номера блоков используются временно, когда осуществляются обращения к барабану, ленте и внешнему оборудованию. Эти номера блоков не появляются в "справочнике блоков". Например, при обращении к магнитных лентах, странице запоминающего устройства на сердечниках временно присваивается номер блока, который опознается схемой, связанной с этим каналом ленты. Когда обращение заканчивается, восстанавливается соответствующий номер блока. Во время обращения к внешнему оборудованию, а также в других случаях, необходимо, чтобы блок хранился в запоминающем устройстве на сердечниках и не передавался бы на барабан. Нужная страница запоминающего устройства "блокируется" установкой разряда в дополнительном запоминающем устройстве. Самообучающаяся программа никогда не выбирает для передачи на барабан страницу, для которой устанавливается этот блокирующий разряд.

3.6.3. ПРОГРАММА УПРАВЛЕНИЯ МАГНИТНОЙ ЛЕНТЫ

Магнитная лента

Лентопротяжный механизм, используемый в вычислительной машине "Атлас", представляет собой Амтекс TM2 (усовершенствованный FR 300), использующий магнитную ленту шириной 2,54 см. На ленте шестнадцать дорожек: двенадцать дорожек информации, две синхронизирующие дорожки и две дорожки, используемые для справочных целей. Информация на лентах записана в виде фиксированных блоков информации по системе предварительно задаваемого адреса. Информация запоминается на ленте блоками по 512 48-разрядных слов вместе с 24-разрядной контрольной суммой с циклическим переносом. Каждому блоку предшествует адрес блока и маркер блока, а оканчивается он маркером блока. Адрес ведущего блока располагается последовательно по ленте, и эффективным является то, что адрес последнего блока всегда равен нулю. Ленты испытываются и размечаются по адресам заранее специальными программами, прежде чем они вводятся в действие, и фиксированная позиция адреса позволяет производить перезапись по выбору и просто пропуск "плохих" мест на ленте. Блоки могут считываться, когда лента движется либо вперед, либо в обратном направлении, но запись возможна только при движении ленты вперед. Двойная головка считывания и записи используется для контроля считыванием при записи на ленту. Когда нет работы, лента останавливается с головкой считывания посередине между блоками информации.

"Атлас" может использовать максимально 32 магнитофона. Каждый магнитофон соединяется с Центральной вычислительной машиной через один из восьми каналов, которые все могут работать одновременно, причем на каждом производится одно считывание, запись или перемотка. Возможно, что каждый магнитофон соединяется с каким-либо одним из пары каналов, при этом переключение находится под управлением организующей программы благодаря разрядам в запоминающем устройстве "V". Ускоренные операции прямой протяжки и обратной перемотки ленты являются автономными и нужен только канал для того, чтобы начать или, если потребуются, закончить их. Передача блока информации из 512 слов между запоминающим устройством на сердечниках и лентой осуществляется через буферное запоминающее устройство на одно слово, причем время ожидания Центральной вычислительной машины в среднем колеблется около 0,5 мксек всякий раз, когда передается слово в запоминающее устройство на сердечниках или из него. Во время передачи странице запоминающего устройства на сердечниках дается особый резервный номер блока, а содержимое регистра адресов страниц восстанавливается в конце передачи.

Организуемые программы запускаются только тогда, когда считываются адреса блока до и после каждого блока и когда останавливается лента. Адрес каждого блока считывается, записывается в запоминающее устройство "V" и устанавливается триггер прерывания работы, вызывая запуск программы прерывания по адресу блока.

Программа прерывания по адресу блока

Эта программа занимается вводом и контролем за передачей отдельного блока между лентой и запоминающим устройством на сердечниках и поиском на ленте адреса определенного блока. Специальные разряды в запоминающем устройстве "V" управляют скоростью и направлением движения ленты, а также началом и окончанием передачи со считыванием или записью. Адреса блоков проверяются от начала и до конца, причем передача с записью не начинается до тех пор, пока адрес ведущего блока ленты не будет считан и проверен. При всех обменах обеспечивается аппаратный контроль, который действует с помощью подпрограмм организующей программы. Образуется 24-разрядная контрольная сумма и проверяется при передаче каждого блока на ленту или с ленты, и, если обнаруживается какое-либо повреждение, устанавливается разряд в запоминающем устройстве "V". Подобным же образом устанавливается разряд в случае сбоя при передаче полного блока из 512 слов. Эти разряды проверяются программой прерывания по адресу блока по окончании каждой передачи. Сбой при контроле по четности либо при считывании из запоминающего устройства на сердечниках, либо при образовании разрядов четности во время передачи в запоминающее устройство на сердечниках, вызывает установку триггера прерывания работы. Если лента останавливается в результате неисправности, это обнаруживается программой прерывания по адресу блока, как особый случай сбоя по адресу блока. Сбой при запуске программы адресов блока (например, сбой при считывании маркеров блоков) обнаруживается с помощью синхронизированной программы прерывания в интервалах 100 мсек. В конечном счете, повреждения магнитофонов, например, повреждения вакуумной системы, устанавливают отдельный триггер прерывания. Нахождение любой из этих ошибок вызывает запуск программ управления лентой, действие которых будет описано ниже.

Организация операций с лентой

Операции с магнитной лентой начинаются запуском подпрограмм организующей программы управления лентой в фиксированном запоминающем устройстве с помощью команд

экстракода основной программы, или, если организующая программа требует операции с лентой для своих собственных целей, — с помощью программ управления экстракодов. Из таблицы в основном запоминающем устройстве, логический номер ленты, используемый в программе, преобразуется в номер действующего механизма, и "команда" ленты вводится в "список" последовательности таких команд, ожидающих выполнения, в основном запоминающем устройстве. Команда ленты может состоять из передачи нескольких блоков информации, из которых какие-нибудь блоки в памяти "блокируются" для того, чтобы предотвратить последовательное их использование до окончания передач. Если какой-либо блок уже входит в передачу, программа, начинающая запрос, останавливается. Подобным же образом, программа останавливается, если список последовательности команд ленты уже заполнен. Если команды, с которыми может быть соединен магнитофон уже заняты передачей или перемоткой, управление ленты возвращает управление основной программе, которая затем освобождается для продолжения операций. Таким образом, программа может потребовать ряд обменов с лентой, не останавливаясь при этом, разрешая, фактически, максимально возможное перекрытие между Центральной вычислительной машиной и магнитофонами во время выполнения программ. Для того, чтобы иметь в распоряжении канал в то время, когда команда ленты вводится в список последовательности, эта команда начинается сразу же записью соответствующих разрядов в запоминающее устройство "V", и записью резервных номеров блоков при передаче с ленты в соответствующие регистры адресов страниц, если команда состоит из передачи со считыванием или записью. Затем управление ленты возвращает управление основной программе или подпрограмме организующей программы.

Используется один составной список последовательности команд ленты для команд, относящихся ко всем магнитофонам, и команды извлекаются из списка последовательности с помощью подпрограмм экстракода организующей программы, запущенных программой прерывания по адресу блока. При считывании адреса предпоследнего блока, входящего в операцию (например, адреса последнего ведущего блока при движении ленты вперед), определяется место следующей операции в этом канале, и если эта операция использует тот же магнитофон, что и данная команда при движении ленты в том же направлении, операция "подготавливается" с помощью запроса какого-нибудь блока, входящего в запоминающее устройство на сердечниках. При считывании адреса последнего блока и успешном завершении контроля, программа прерывания по адресу блока начинает немедленно следующую операцию, если она уже подготовлена, избегая, таким образом, останова ленты, если это возможно. Если не подготовлена никакая операция, программа прерывания останавливает ленту установкой разряда в запоминающем устройстве "V", и происходит дальнейшее "прерывание по адресу блока", когда лен-

та останавливается, и канал может принимать дальнейшие команды. Это прерывание запускает подпрограмму экстракода организующей программы, которая извлекает следующую команду для этого канала из списка последовательности на ленте, и цикл действий повторяется до тех пор, пока не останется ни одной команды для этого канала. Когда заканчивается каждая передача, любая основная программа, остановленная при обращении к этому блоку запоминающего устройства, освобождается для продолжения работы.

Исключением из описанного выше процесса является случай, когда требуется долгое движение (свыше 200 блоков информации) или перемотка. В этом случае движение происходит на большой скорости при запрещении прерывания работы по адресам блоков, а тем временем этот канал может быть использован для управления другим магнитофоном. Долгое движение ленты заканчивается контролем в течение короткого отрезка времени и в соответствующий момент, и запускается управление ленты от синхронизированной программы прерывания. Затем магнитофон переводится обратно "на канал", и скорость становится нормальной. Когда считывание адреса блока происходит правильно, поиск адреса продолжается обычным путем.

Блок названий

Первый блок информации на каждой магнитной ленте резервируется для использования организующей программой, и выборка информации из этого блока с помощью основной программы происходит только по специальным командам. Этот блок содержит название ленты или указание о том, что лента свободна. Когда магнитные ленты запрашиваются организующей программой или основной программой, организующая программа печатает команды для оператора для того, чтобы заправить названную ленту и включить магнитофон, на который она заправлена. Кнопка включения каждого магнитофона (см. раздел 5) соединена с разрядом запоминающего устройства "V", и эти разряды опрашиваются организующей программой управления каждую одну секунду. Когда обнаруживается состояние "включено", вводится управление ленты для считывания первого блока в ленте. Затем название ленты проверяется на соответствие предполагаемому названию. Таким образом, подтверждается наличие правильной ленты, и, кроме того, лента, несущая это название, связывается с данным магнитофоном. Ввиду того, что программист приписывает логический номер ленты ленте, несущей данное название, этот логический номер ленты, используемый в командах экстракода, может быть преобразован организующей программой в действительный номер данного магнитофона. В первый блок на каждой ленте вводятся другие данные управления, включая номер ленты системы и количество блоков информации на ленте. Особые подпрограммы организующей программы позволяют машине "Атлас" считывать ленты, производимые для вычислитель-

ной машины "Орион" фирмой Ферранти, которая использует такие же лентопротяжные механизмы, но может записывать на ленту блоки переменной длины. Эти ленты на машине "Атлас" различаются по маркеру, записанному в блоке названий.

Повреждения магнитной ленты

Все неисправности, обнаруженные программами прерывания, заставляют программу прерывания по адресу блока останавливать ленту, по возможности в конце данного блока и затем запускать программы управления контролем ленты. Если ленту нельзя остановить, она отключается и запускаются контрольные подпрограммы управления лентой. Эти подпрограммы представляют собой подпрограммы экстракода организующей программы составленные так, чтобы уменьшить непосредственное влияние отдельных ошибок системы лент на Центральную вычислительную машину, сообщать инженерам по эксплуатации о любых неисправностях и, по возможности, определять источник неисправности. Как пример действий, осуществляемых контрольными подпрограммами, предположим, что в процессе считывания блока с ленты в запоминающее устройство на сердечниках обнаруживается ошибка по контрольной сумме. Контрольные подпрограммы ленты делают до двух дальнейших попыток считать блок. Если какая-нибудь из этих попыток достигнет цели, то после осведомления об этом инженеров, снова вводится обычное управление лентой. Повторение ошибки может быть вызвано лентой или лентопротяженным механизмом. Чтобы различить эти причины, лента перематывается и делается попытка считать первый блок. Если это проходит успешно, то указывается на ошибку ленты, и делается попытка считать подозреваемый блок с уменьшенным уровнем смещения. Неисправность является причиной того, что механизм магнитофона отключается, и программа, использующая ленту, приостанавливается. Если "повторное считывание" является успешным, эта лента переписывается на свободную ленту, и операторы информируются о переадресации ошибочной ленты с пропуском того блока, в котором имеется ошибка. Если при перематке ленты первый блок не может быть правильно считан, подозревается неисправность в магнитофоне, и оператору дается команда о переустановке ленты на другой магнитофон. Другие неисправности проверяются таким же способом, и повсюду оператору и инженерам сообщается о любых обнаруживаемых ошибках. Предусматриваются меры для программы, использующей ленту, с целью "захватить" постоянные ошибки ленты и посредством этого определять действие, пригодное для данной задачи, которое может быть более прямым и эффективным, чем стандартное действие управления.

Адресация новых лент и переадресация ошибочных лент выполняются вычислительной машиной с помощью подпрограмм организующей программы, вызываемых оператором. Магнитофон переключается на "способ адресации", который запре-

дает передачи в запоминающее устройство на сердечниках и из него, позволяет производить запись из вычислительной машины на контрольные дорожки и в адреса блоков на ленте и приводит в действие синхронизирующий механизм для разметки адресов блоков. Когда новая лента размечается по адресам, адреса записываются на ленте последовательно и область между адресами первого и последнего блока контролируется записью единиц во всех цифровых позициях и проверкой на себя при обратном считывании. Любой блок, вызывающий собой, стирается, и лента размечается соответствующим образом. По окончании записывается специальный адрес блока, чтобы показать "конец ленты", и вся лента тогда проверяется с помощью считывания в обратном направлении. Всякие собой вызывают ввод программы переадресации. В конечном счете магнитофон возвращается к "нормальному способу", записывается блок названий, содержащий количество блоков на ленте, номер ленты и название "Свободно", и лента снова пригодна для использования. Лента, имеющая ошибочные блоки, переадресовывается с пропуском таких блоков, с помощью ввода программы переадресации со списком ошибочных блоков. Ошибочные блоки стираются, а оставшиеся блоки последовательно перерематываются, и лента проверяется как при адресации новой ленты.

3.6.4. ВНЕШНИЕ УСТРОЙСТВА

Прерывания работы внешнего оборудования

К вычислительной машине "Атлас" может быть присоединено большое количество разнообразных внешних устройств. Однако количество электронных схем, связанных с каждым устройством, сводится к минимуму, а для контроля этих устройств и в качестве буферной памяти большой емкостью используются высокая скорость и возможности прерывания "Атласа".

Таким образом, устройства для считывания с бумажной перфоленты, которые работают со скоростью 300 знаков в секунду, устанавливают триггер прерывания всякий раз, когда появляется новый знак. (Знаки могут быть либо 5-, либо 7-разрядными в зависимости от двух возможных значений ширины считываемой ленты). Подобным же образом перфораторы на бумажной ленте и телетайпы, печатающие информацию для операторов вычислительной машины, вызывают прерывание работы всякий раз, когда они готовы получить новый знак. Это оборудование работает соответственно со скоростью 110 и 10 знаков в секунду.

Устройства считывания с перфокарт считывают 600 карт в минуту, столбец за столбцом, и прерывают работу вычислительной машины для каждого столбца. Перфораторы со скоростью 100 карт в минуту пробивают по рядам и прерывают работу машины для каждого ряда.

Печатающие устройства, имеющие 120 печатных колес, несущие по 50 различных знаков, вызывают прерывание по мере того, как каждый знак приближается к печатающей позиции, так что вычислительная машина может подготавливать молоточки для этих колес там, где этот знак должен быть напечатан. Следовательно, за оборот происходит 50 прерываний, или каждое одно прерывание происходит через 1,5 мсек.

Вся информация, получаемая из этого внешнего оборудования или посылаемая в него, поступает, таким образом, через особые разрядные позиции в запоминающее устройство "V". Например, имеется по 7 таких разрядов на каждое устройство считывания с перфоленты и по 120 — на каждое печатающее устройство вместе с несколько большим количеством разрядов для сигналов управления.

Большинство прерываний может осуществляться просто с помощью программы прерываний для данного типа оборудования. Таким образом, программа прерывания для устройства считывания с бумажной перфоленты обычно должна просто обращаться к таблице знаков, чтобы применить преобразование кода и контроль по четности и обнаружить последние знаки, и если все хорошо, запомнить этот новый знак в следующей позиции запоминающего устройства.

Не предполагается, что программы прерывания для печатающих устройств и перфораторов делали бы преобразование из знаковых кодов в двоичный ряд. Это делается с помощью подпрограммы экстракода организующей программы до того, как начнется пробивка карты или печатание строки. Программы для карт, однако, усложняются узлами контроля считывания. Пробивка карт проверяется на один цикл карты позже, а считывание проверяется позже на время, соответствующее вводу трех столбцов. Программы прерывания применяют эти проверки, и в случае сбоя вводятся подпрограмма экстракода организующей программы.

Программа прерывания печатающего устройства считает количество своих прерываний, чтобы опознать знак, который печатается в данный момент. Проверка обеспечивается один раз при каждом обороте, когда отметка данной величины на валике печатающего устройства заставляет сигнальный разряд появляться в запоминающем устройстве "V". Этот подсчет производится во время протяжки бумаги между строками, что занимает несколько мсек.

Обслуживание с помощью оператора

Всякий раз, когда оборудование нуждается во внимании, оно "отсоединяется" от вычислительной машины. В этом состоянии, которое указывается лампочкой на оборудовании и соответствующим разрядом в запоминающем устройстве "V", это устройство автоматически останавливается и не может быть приведено в действие вычислительной машиной.

Оператор может включить или отключать оборудование посредством двух кнопок "включено" и "выключено". Обо-

рудование может также отключаться вычислительной машиной с помощью записи соответствующего разряда в запоминающем устройстве "V", но вычислительная машина не может сама включать оборудование.

Кнопки "включено" и "выключено" сами не вызывают прерывания работы Центральной вычислительной машины. Взамен этого в запоминающем устройстве "V" каждую секунду проверяются разряды "включено" (эта подпрограмма приводится в действие синхронизирующими прерываниями), и любое изменение в состоянии приводит в действие соответствующую подпрограмму экстракода организующей программы. Выключение устройства не является запретом к его прерываниям, так что если оператор выключает перфоратор карт в середине цикла, чтобы пополнить магазин или вынуть карты из механического укладчика, то цикл заканчивается правильно.

Есть также и другие специальные управления для отдельных устройств, например, кнопка вывода у машин для перфорирования карт и селекторный переключатель с 5-й на 7-дорожечную перфоленту.

Большинство устройств имеют детекторы, которые показывают, когда запас карт или бумаги исчерпывается или они проходят неправильно. Это соответствует разрядам запоминающего устройства "V", которые считываются соответствующей подпрограммой экстракода организующей программы. Однако устройства считывания с бумажной перфоленты не имеют такого детектора, маловероятный случай, что перфолента проходит полностью через считывающее устройство (из-за отсутствия оконечных знаков), воспринимается вычислительной машиной просто как сбой при ожидании следующего знака в пределах нормального промежутка времени. Это условие обнаруживается односекундной программой.

Организация запоминающего устройства для ввода и вывода информации

Вообще информация ввода превращается в стандартный шестиразрядный внутренний код знака с помощью соответствующей программы прерывания и в запоминающем устройстве занимает 8 знаков на слово. (Исключение бывает в случае наличия устройств считывания с перфокарты, когда считываются карты, перфорированные не в стандартном коде, в этом случае 12 разрядов из одной колонки просто записываются в запоминающее устройство и занимают две знаковых позиции. Подобный случай происходит при 7-дорожечной перфоленте, когда для передачи используются 7 разрядов информации без контроля по четности. Такая информация отличается предупреждающими знаками как в самих устройствах ввода, так и в запоминающем устройстве).

Некоторое рабочее пространство для организующей программы в запоминающем устройстве на сердечниках не учитывается для получения информации ввода от программ прерывания работы и делится между различными внешними устройствами ввода. Объем этого пространства зависит

от количества и типа участвующих внешних устройств; первые две машины "Атлас" будут нормально использовать один блок информации (512 слов). Этот блок будет разблокироваться на странице запоминающего устройства на сердечниках всякий раз, когда работает внешнее устройство ввода (т.е. большую часть времени).

Поскольку каждое устройство ввода заполняет свою часть этого блока, то информация переписывается подпрограммой экстракода организующей программы на другой блок, предназначенный исключительно для этого устройства. Эти операции переписи являются достаточно редкими, чтобы была необходимость оставлять этот блок в запоминающем устройстве на сердечниках; в действительности это подлежит такой же обработке, как и для основных программ, с помощью программы обмена с барабаном и может быть введено в барабан и получено обратно снова для следующей операции перезаписи. Таким образом, только одна страница запоминающего устройства на сердечниках используется полное время во время операций ввода, но тем не менее каждый поток ввода находит свой путь в отдельный ряд блоков в запоминающем устройстве.

Страница, которая делится между внешними устройствами ввода, подразделяется так, чтобы уменьшить до минимума число случаев, при которых информация должна быть переписана на другие блоки; оказывается необходимо, чтобы пространство для каждого устройства было приблизительно пропорционально квадратному корню из скорости подачи информации.

Подобно этому информация, предназначенная для вывода, находится на общей странице вывода, подразделенная для различных устройств вывода, и берется оттуда, по требованию, программой прерывания работы. Программы прерывания для телепринтеров и перфораторов ленты делают необходимым преобразование из внутреннего кода знака, используемого этим устройством. Как только информация для отдельного устройства закончится, начинает работать подпрограмма экстракода организующей программы для переписывания свежей информации на общую страницу вывода. И снова страница подразделяется приблизительно пропорционально квадратному корню из скорости подачи информации.

Что касается перфораторов и печатающих устройств, для которых программы прерывания требуют информацию, расположенную в рядах разрядов, то необходима дальнейшая ступень перевода. В этих случаях, при окончании перфокарты или строки, внутренние 6-разрядные знаки преобразовываются с помощью подпрограммы экстракода организующей программы в "отображение" карты или строки также в блоке вывода. В действительности, желаемый объем рабочего пространства в качестве буферного выходного объема несколько превышает один блок, и запасная емкость вспомогательного запоминающего устройства должна быть использована для его увеличения.

3.6.5. РАБОТА ОРГАНИЗУЮЩЕЙ ПРОГРАММЫ ПРИ ВВОДЕ И ВЫВОДЕ

Ввод

Большая скорость вычисления машины "Атлас" и использование внешних устройств для многократного ввода и вывода дает возможность вычислительной машине обрабатывать большое количество разнообразных задач. В этот диапазон входят маленькие задачи, для которых не существует данных вне самой программы, и большие задачи, требующие нескольких групп данных, возможно приходящих из различных источников. Другие вопросы ввода могут состоять из исправлений к программам или требований выполнить уже введенные программы. Несколько подобных вопросов может быть представлено на рассмотрение вместе в одной пачке карт или на одной бобине перфоленты. Все должно быть надлежащим образом установлено для вычислительной машины.

Для того, чтобы систематизировать эту задачу определений, вводится понятие "документ". "Документ" — самостоятельный объем информации ввода, который вводится в вычислительную машину последовательно через один канал ввода. Каждый "документ" несет соответствующую определяющую информацию (см. ниже), и организующая программа содержит в главном запоминающем устройстве список "документов", по мере того как они принимаются в запоминающее устройство с помощью программы ввода, и список задач, для которых ожидаются последующие "документы".

Задача может потребовать несколько "документов" и только когда все они будут введены, можно начать вычисление. Поэтому организующая программа проверяет поступление "документов" для каждой задачи; когда они собраны, запускается программа, определяющая план решения задачи (см. ниже).

Обычно главное запоминающее устройство и запоминающее устройство на барабанах в вычислительной машине, вероятно, не удовлетворяют требованию хранения всех "документов", которые находятся в ожидании своего использования. Поэтому блоки информации ввода по мере получения переписываются на магнитную ленту, принадлежащую организующей программе, которая называется "лентой системы ввода". Отсюда, если организующей программе необходимо стирать их в главном запоминающем устройстве, то они могут быть введены еще раз с ленты системы ввода, когда задача готова для вычисления.

Таким образом, лента системы ввода работает как буфер большого диапазона и, действительно, она играет роль такую же, как и лента системы ввода в более стандартных системах.

Различия здесь заключаются в том, что лента подготавливается самой вычислительной машиной вместо того, чтобы эта лента готовилась каким-либо устройством независимо от машины и что не существует протяжки ленты или ручного управления, которые бы требовались после ввода исходных "документов" — а это важно для системы, спроектированной для обработки многих разнообразных задач.

Эта полная буферная система для документов ввода называется "источник ввода". О "документах", ожидающих следующих "документов", до того как они могут быть использованы, говорят, что они находятся в "источнике ввода А"; полный набор документов для задач образует "источник ввода В". Обычно "документы", будучи введенными в источник ввода В, должны быть считаны с ленты системы ввода обратно в главное запоминающее устройство так, чтобы они были готовы для вычисления; однако часто они уже бывают в источнике ввода А в главном запоминающем устройстве, так что требуется только внести исправления в "справочник блоков" в памяти.

Результат этой организации работы таков, что одна и та же лента используется как для последовательной записи блоков ввода, так и для обратного считывания предварительно записанных блоков, чтобы обновлять отдельные документы, когда они требуются. Поэтому, к ленте будут частые обращения, на протяжении нескольких футов ее длины, хотя она и будет постепенно продвигаться вперед. Длины этих просматриваемых участков ленты связаны с объемом главного запоминающего устройства, занятого источником ввода А. Например, поскольку эти участки ленты не превосходят 80 футов, т.е. около 24,5 м (130 блоков информации), то времени ожидания для записи новых блоков будет оставаться меньше, чем для ввода трех блоков с устройства считывания с перфокарт, так что необходимо, чтобы сравнительно малый объем основного запоминающего устройства был занят источником ввода А. Для обеспечения этого просматриваемые участки ленты поддерживаются в разумных пределах, а любые "документы", оставленные на ленте системы ввода в течение такого продолжительного времени, что они приближаются к пределу просматриваемой области, переписываются на ленту системы останова работы (см. ниже). Если количество таких "документов" становится большим, то оператор вычислительной машины предупреждается, чтобы он уменьшил подачу "документов" через внешнее устройство ввода.

Вывод

Центральная вычислительная машина может давать информацию на вывод при значительно большей скорости, чем могут получать их внешние устройства и "источник вывода" используется по способу, аналогичному источнику ввода.

Этот источник использует "ленту системы вывода" в виде буфера для всего объема данных.

Вывод для всех внешних выводных устройств проводится на той же ленте, разбитой на секции, которые подразделены так, что содержание одной секции будет занимать все работающие в данное время внешние устройства в один и тот же отрезок времени. Таким образом, если например, неожиданно появляется информация вывода для отдельного внешнего устройства, то она размещается на ленте системы вывода, оставляя запасные блоки, которые должны быть заполнены позже информацией вывода для других внешних устройств (это возможно, потому что машина "Атлас" использует заранее размеченную по адресам ленту). Таким образом, возвращение информации с ленты в "источник вывода В", как требуется различными внешними устройствами, просто влечет за собой считывание полной секции информации с ленты.

Кроме того, существует предел количества информации, которая может быть полезно размещена в качестве буфера на ленте вывода, вследствие времени, требуемого для протяжки ленты взад и вперед между областями записи и считывания; этот предел зависит от объема, занимаемого в главном запоминающем устройстве источником вывода В. Подпрограмма экстракода организующей программы контролирует количество информации, остающейся в источнике вывода В для каждого устройства, и связывает его с длиной обозреваемого участка ленты для того, чтобы решить, когда начать перематывать ленту назад для следующей операции считывания. Если объем информации вывода, вырабатываемый основной программой, становится слишком большим, то некоторая ее часть помещается на ленту останова работы (см. ниже) или же программа останавливается.

Разгрузочная лента

Ленты системы ввода и вывода служат в основном для расширения объема памяти главного запоминающего устройства вычислительной машины. Говоря шире, "документы" подаются в вычислительную машину, выполняется программа и производится вывод информации. Тот факт, что информация ввода и вывода обычно находится некоторое время на магнитной ленте, является, в некотором смысле, случайностью. Однако наличие буфера для информации ввода и вывода является непрерывной и специальной потребностью, поэтому был разработан частный случай использования этих лент и была написана специальная подпрограмма экстракода организующей программы для управления этими лентами.

Если требования к памяти превышают емкости главного запоминающего устройства и лент ввода и вывода, то используется отдельная магнитная лента, разгрузочная лента, для хранения той информации, которая не требуется немедленно. Эта лента может быть введена для различных целей. Решение задачи может быть остановлено и за-

дача записана временно на разгрузочную ленту, если требуется, чтобы другие задачи заполнили область памяти, предназначенную для информации вывода (источник вывода), или наоборот, если ее собственный вывод не может быть размещен в "источник вывода". Так же, как уже описывалось, источники ввода и вывода могут сами "переполняться" и заполнять разгрузочную ленту. Эта лента систематически не используется, а применяется в случае крайней необходимости. Однако, если нужно, ленты системы ввода и вывода могут обходиться без системы, подобной этой, уменьшая тем самым объем памяти для информации ввода и вывода и увеличивая нагрузку на разгрузочную ленту машины. В крайнем случае можно обойтись и без самой разгрузочной ленты за счет снижения эффективности системы.

Заголовки и названия

Каждому вводимому документу предшествует опознавательная информация. Она состоит из двух напечатанных строк, образующих соответственно заголовок и название.

Заголовок указывает на то, какой тип документа следует. Наиболее общие заголовки следующие:

	Компилирующая программа, за которым следует название "языка программ", что обозначает, что документ является программой в установленном языке;
Данные	- означает, что документ является данными, затребованными дополнительной программой;
Задача	- означает, что документ является требованием для вычислительной машины производить вычисление задачи и дается несколько соответствующих данных о ней.

Последний вид документа называется "описанием задачи". Он дает, например, список всех других документов, которые требуются для задачи, список выданных потоков информации вывода, любых требуемых магнитных лент и требуемых верхних пределов объема запоминающего устройства и требуемого времени вычисления. Многие из этих подробностей необязательны; например, если не ссылаться на объем запоминающего устройства и время вычисления, то будет сделано стандартное допущение.

Например, если программа работает по двум документам данных, которые обозначаются как "данные 1" и "данные 2", то описание задачи будет содержать:

Вывод

1 за которым следует название данных 1,

2 за которым следует название данных 2.

Программа появится в этом списке как "данные 0". Или же описание задачи может быть соединено с программой, образуя один составной документ. Так обычно происходит с маленькими задачами.

Каждый поток информации вывода может быть предназначен отдельному внешнему устройству или типу внешнего устройства, или можно допустить, что он может быть выведен на любом выходном устройстве. Объем информации вывода в каждом потоке может быть также определен.

Например, описание задачи может состоять из:

Вывод

1 построчно-печатающее устройство, 20 блоков,
2 перфокарты,
3 любое.

Каждая магнитная лента, используемая программой, определяется номером внутри программы, и описание задачи содержит список этих номеров с названием, которое появляется в блоке 0 каждой ленты для ее определения, например:

Лента

1 цилиндр потенциального поля (204) TRU5.

Если требуется новая лента, то свободная лента должна быть заправлена, которую затем программа может принять и дать новое название. Это указывается следующим образом:

Лента свободна

2 результаты Монте-Карло K49-REAC-OR4.

Требование для заправки лент операторами задается организующей программой, работающей по информации в описаниях задач. И, наконец, окончание "документа" обозначается

* * *

а если это также и окончание перфоленты или пачки перфокарт, то за этим обозначением следует буква Z. После того как вычислительная машина прочтет это обозначение, она выключает устройства.

Регистрация и оценка машинного времени

Когда решение задач закончено, то организующей программой накапливаются различные вопросы информации по работе вычислительной системы. Накапливаются такие вопросы, как изменение числа программ и количество обменов с

барабаном, а также для каждой задачи количество команд, время, потраченное на ввод и вывод и использование магнитных лент. Эти вопросы печатаются в группах для обеспечения операторов записью характеристик работы вычислительной машины, они также необходимы для оценки стоимости работы на машине.

Метод подсчитывания стоимости может значительно изменяться в зависимости от различных установок, но единственная желаемая черта любого метода заключается в том, чтобы оценка стоимости работы программы изменялась незначительно от одной работы до другой. Трудность заключается в том, что количество обменов с барабанами, требуемое в программе, может значительно изменяться в зависимости от объема запоминающего устройства на сердечниках, который используется в одно и то же время для магнитной ленты и обменов с внешними устройствами. Один метод оценки стоимости работы без учета этого изменения заключается в том, чтобы не делать оценки обменов с барабанами, но основывается на оценке стоимости времени вычисления по количеству команд, используемых в программе. Это, однако, не побуждает программиста составлять программу так, чтобы уменьшить в ней количество обменов с барабаном, и в конечном итоге может привести к разработке более сложных схем. Оценка стоимости использования внешних устройств для ввода и вывода может быть проведена на основании объема информации ввода и вывода. Что касается магнитных лент, то оценка может быть основана на времени, в течение которого включен магнитофон, при этом должно быть принято в расчет время, когда программа готова к работе, но остановлена программой более старшей по своему значению. Вся эта информация предназначается для программы управления экстракодами, ответственной за стоимость решения задач.

Методы работы вычислительной системы

Обычный метод работы вычислительной машины — это обработка документов, подаваемых в любое внешнее устройство в любом порядке, хотя обычно связанные друг с другом документы должны подаваться в одно и то же время. Названия и описания задачи позволяют организующей программе собирать и выполнять сложные программы, а информация вывода распределяется по всем имеющимся внешним устройствам. Обычно программы составляются и вычисляются в том же порядке, в каком завершается и ввод, но организующая программа может изменить этот порядок в зависимости от загрузки различных частей системы. Например, задача, требующая магнитофоны, которые уже находятся в использовании, может быть пропущена в пользу задачи, использующей свободное внешнее устройство вывода; задача, которая вычисляется долгое время, может быть временно приостановлена для того, чтобы увеличить загрузку внешних устройств вывода. Этими и подобными способами подпрограмма экстракода организующей программы, ответ-

ственная за работу по плану, пытается поддержать, по возможности, наиболее полную загруженность внешних устройств вывода, магнитофонов и Центральной вычислительной машины.

"Документы" могут быть также поданы в вычислительную машину с магнитных лент; это могут быть либо ранее использовавшиеся ленты системы ввода или "библиотечные" ленты, или ленты "стандартные", которые используются часто, и на которых хранятся программы. Такие документы рассматриваются в качестве составной части "источника ввода В" и по требованию считываются в главное запоминающее устройство. Другой метод работы может заключаться в использовании вычислительной машины для переписывания документов на "частную" магнитную ленту, а не на ленту системы ввода, и позже подавать в вычислительную машину последовательность задач с этой ленты. Аналогично этому вывод может быть накоплен на "частную" магнитную ленту и позже выведен через вычислительную машину на одно или более внешних устройств. Подпрограммы, являющиеся частью организующей программы, могут выполнять подобные стандартные операции "переписывания".

Для главного оператора предусмотрено также, что он может модифицировать систему по-разному: например, первенство может быть предоставлено какой-нибудь отдельной задаче, или внешнее устройство может быть выведено из общего использования для выполнения особой задачи. Может быть, например, установлено состояние "изолированной работы", при этом резервируется отдельное устройство вывода, предназначенное для задач, вводимых с отдельного устройства ввода.

ЗАКЛЮЧЕНИЕ

Возможно, организующая программа машины "Атлас" является наиболее совершенным примером из всех до сих пор встречавшихся программ, в которые входит много параллельных, тесно связанных между собой работ. Хотя большая часть программы и была уже составлена ко времени написания статьи, все еще существует несколько деталей, которые должны быть еще выяснены и несомненно многие изменения для того, чтобы удовлетворить условиям, имеющим место в различных установках. Поэтому полная структура программы имеет главное значение: только если эта структура отвечает всем требованиям, прочна и систематична, то можно будет удовлетворительно закончить составление программы, и произвести необходимые изменения по мере их возникновения.

Структура, описанная в этой статье, подпрограммы прерывания работы и программы экстракода организующей программы, управляемые координирующей программой, оказалась вполне удовлетворительной как основа для каждой задачи управления, которая до сих пор была предусмотрена, и можно предполагать, что все изменения, ко-

торые могут потребоваться, будут приспособлены к этой структуре.

Нет сомнения, что успешное применение этой организующей программы для очень быстрой вычислительной машины обуславливается в основном благодаря некоторым свойствам аппаратуры "Атласа", в частности, условиям для защиты программ от взаимного вмешательства и регистров адресов страниц. Способ, которым последние позволяют информации быстро обновляться без необходимости повторно восстанавливать программы, придает организующей программе совершенно новую степень свободы. Возможные применения этой схемы оказываются такими обширными, что задача использования такой большой вычислительной машины эффективно без этой возможности кажется, путем сравнения, совершенно невозможной.

Раздел 4 - ГЛАВНОЕ ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО, И СХЕМА ОБРАЩЕНИЯ К БАРАБАНУ

4.1. ОБЩИЕ ПРИНЦИПЫ ЗУ ОДНОГО УРОВНЯ

Нормально программист составляет команды, обращаясь непосредственно к ЗУ на сердечниках или барабанах. Нельзя передавать информацию из ЗУ на барабанах в накопитель и В-регистры, не поместив ее предварительно в ЗУ на сердечниках; все передачи на барабан, необходимые для этого, обеспечиваются машинными устройствами и фиксированной программой памяти.

Так как все обмены с барабанами осуществляются "блоками", информация по 512 48-разрядных слов, ЗУ на сердечниках можно считать состоящим из "р" страниц по 512 слов каждая. Подобным же образом сектор на барабанах можно рассматривать состоящей из s "секторов" по 512 слов. Общая емкость памяти машины таким образом составляет b блоков информации, состоящих каждый из 512 слов, где $b = r \cdot s$. С каждой страницей ЗУ на сердечниках связан "адрес страницы" или "регистр совпадения", состоящий из 12 триггеров. Одиннадцать из них содержат номер блока информации, находящейся сейчас на соответствующей странице. Двенадцатый триггер - это разряд "блокировки", который устанавливается в "1", когда данный блок не может быть использован в основной программе, например во время обращения к этому блоку барабанов или внешних устройств (оставшиеся 10 разрядов адресов не должны запоминаться в этих регистрах, т.к. они указывают только место слова внутри страницы или блока). Эти регистры адреса страницы являются частью запоминающего устройства "v" на сердечниках. В этом ЗУ "v" есть также разряд "использования" для каждой страницы ЗУ на сердечниках.

Когда должна выполняться команда, которая ссылается на адрес в основном ЗУ, этот адрес, предварительно В-модифицированный, сравнивается с каждым из "р" адресов страниц, чтобы определить, находится ли уже слово с этим адресом в ЗУ на сердечниках. Если оно там находится, ко-

манда выполняется как обычно. Следует заметить, что если адрес, указанный в команде, есть $m.l$ (где l - строка внутри блока в основном ЗУ), адрес этого слова в ЗУ на сердечниках будет $s.R.$, причем "m" и "с" обычно не совпадают. Адрес $s.R.$ является таким образом тем адресом, содержимое которого обрабатывает машина.

Если указанная ячейка не находится в ЗУ на сердечниках, то номер, в данный момент находящийся в устройстве управления, запоминается, и начинает выполняться специальная программа обращения к барабану, находящаяся в фиксированном ЗУ. С этой программой связано ($p+s$) слово во вспомогательном ЗУ, т.е. указатели страницы и сектора, дающие номера блоков информации, находящихся на каждой странице и секторе. Когда блок информации передается в некоторый сектор, устанавливается в "I" разряд в соответствующем адресе указателя сектора, чтобы указать, что в этом секторе сейчас записана необходимая информация. Когда данный блок считывается в ЗУ на сердечниках, этот разряд очищается.

Программы обмена с барабанами должны (а) подготовить страницу в ЗУ на сердечниках для поступления новой информации, записав имеющуюся там информацию на одну из страниц ЗУ на магнитных барабанах; (в) считать требуемую информацию из ЗУ на магнитных барабанах на эту страницу.

Решение о том, на какую страницу необходимо произвести запись, принимает самообучающаяся программа обмена с барабанами (см. раздел 4.2.) с помощью указанных выше разрядов "использования". Вопрос, на какой сектор нужно записать, решается путем считывания адреса сектора с барабана и определения по информации, содержащейся в списке секторов, первого сектора, не содержащего полезной информации, на который можно произвести запись.

После того, как нужный блок передан из соответствующего сектора на страницу, сделанную для него доступной в ЗУ на сердечниках, производятся необходимые изменения в списке страниц и секторов; регистр адреса страницы, связанный с той страницей в ЗУ на сердечниках, которая сейчас содержит данный блок информации, также меняется так, чтобы он содержал адрес блока. Эти регистры устанавливаются программой для барабанов, хранящейся в постоянном ЗУ.

Оказалось, что для машины, в которой ЗУ на сердечниках содержит 32 страницы, выгодно сделать 31 страницу доступной для программистов, а одну страницу всегда иметь свободной, чтобы туда можно было передать очередной блок информации из ЗУ на магнитных барабанах. Таким образом, когда достигнуто несовпадение с регистрами страниц, требуемый блок немедленно переносится в ЗУ на сердечниках и, пока это производится, можно решить, какую страницу освободить для следующей передачи. Передача этой страницы на барабан совершается в ходе выполнения главной программы. После конца передачи информации с барабана в ЗУ на сердечниках, регистр управления возвращается в исходное состояние и команда, которая выз-

вала перерыв в выполнении программы, снова начинает выполняться.

4.2. САМООБУЧАЮЩАЯСЯ ПРОГРАММА ОБРАЩЕНИЯ К БАРАБАНАМ

Эта программа используется в качестве подпрограммы программы обмена с барабанами. Ее цель - указать, к какому блоку информации в ЗУ на сердечниках имеется наименьшая вероятность обращения в программе и потому какой блок должен быть переписан на барабан, когда требуется место в ЗУ на сердечниках для записи нового блока информации с барабана или с магнитной ленты.

Чтобы позволить программе принять это решение, были созданы разряды "использования". Эти разряды, которые так же, как и регистры адреса страницы входят в ЗУ "v" на сердечниках. На каждую страницу в ЗУ на сердечниках есть один разряд использования, связанный с ней. Разряд использования для некоторой страницы устанавливается в "I" (операция "или"), если делается ссылка на блок информации, записанный сейчас в этой странице.

Еще одно машинное устройство, используемое самообучающейся программой - это "прерыватель счета команд". В ЗУ "v" внешних устройств есть счетчик команд (см. раздел 13.6. Тип 12а). Число в этом регистре увеличивается на единицу, когда заканчивается выполнение очередной команды, причем, если используется главное управление или управление экстракодами, то через каждые 1024 команды устанавливается в "I" соответствующий "сигнальный триггер", который приводит к автоматическому прерыванию работы.

Каждый раз, как происходит такое прерывание работы, разряды использования считываются с ЗУ "v" на сердечниках и заносятся в список, который содержится во вспомогательном ЗУ. Регистры, содержащие разряды использования, автоматически очищаются, когда с них считывается информация. С каждым блоком информации самообучающаяся программа связывает три параметра t , t' и T . Назначение этих параметров в том, чтобы контролировать частоту, с которой обращаются к каждому блоку информации.

Если разряд использования некоторой страницы занят единицей, когда он считывается с ЗУ "v", это указывает, что данная страница хотя бы раз упоминалась в командах со времени последнего прерывания счета команд. Наоборот, если разряд использования содержит нуль, эта страница не упоминалась с момента такого прерывания. Три указанных параметра означают: t - длительность периода, в течение которого не используется блок, находящийся на соответствующей странице ЗУ на сердечниках, т.е. число прерывания счета команд, которые произошли с момента последнего случая использования данного блока.

T - длительность предыдущего (перед последним использованием) периода, в течение которого блок не использовался; если блок был записан на барабан, сюда включает-

ся время его пребывания на барабанах.

t' - начало данного периода неиспользования блока, который записан на барабан. Эта информация сохраняется, когда блок считывается обратно в ЗУ на сердечниках; таким образом можно найти T , зная t и число прерываний от записи до обратного считывания блока.

Параметры t и T меняются каждый раз, когда происходит прерывание из-за "несовпадения" (исключение представляет случай, когда имеется пустая страница, доступная для следующей передачи с барабана; в этом случае список разрядов использования остается неизменным). Эти два параметра сохраняются только для блоков, находящихся в это время в ЗУ на сердечниках. Параметр t' устанавливается только тогда, когда блок записывается на барабан.

Эти параметры используются в следующих правилах для решения того, какой блок должен быть записан на барабан.

1) Проверить, является ли какая-либо страница в ЗУ на сердечниках все еще пустой: в этом случае не требуется передачи информации на барабан.

2) Если пустых страниц нет, записать страницу, для которой величина $(T - t)$ отрицательная и имеет наибольшую величину. (В этом случае и подобных ему случаях, если два или более блока имеют одну и ту же величину, записывается тот, у которого номер страницы больше).

Неравенство $t > T$ означает, что блок в ЗУ на сердечниках оставался в бездействии дольше, чем он оставался в бездействии в последний раз.

3) Если нет страницы, для которой $T - t < 0$, выбрать страницу, для которой: а) $t \neq 0$ и б) $T - t$ имеет наибольшую положительную величину. $t \neq 0$ означает, что блок не использовался со времени последнего обмена.

4) Если все $t = 0$, применять правило 3 б, которое теперь таково: T - имеет наибольшую положительную величину, т.е. передается блок, который имеет тенденцию дольше всего оставаться неиспользованным.

4.3. КОМАНДЫ ЭКСТРАКОДА ОБРАЩЕНИЯ

К БАРАБАНАУ

Можно предвидеть, что большинство программистов будет использовать сердечники и барабаны в качестве ЗУ одного уровня, как это указано в разделе 4.1. Однако в некоторых ситуациях бывает полезно иметь возможность указать, что данный блок информации либо должен быть в ЗУ на сердечниках, либо может быть записан в ЗУ на барабанах. Для этого созданы следующие команды экстракода для обмена с барабанами.

1) Считать блок "b" в первую половину, или во вторую половину, или в любую половину ЗУ на сердечниках.

Если нет подходящей пустой страницы, блок информации, находящийся сейчас в ЗУ на сердечниках, записывается на барабан. Если блок уже находится в указанной половине ЗУ на сердечниках, ничего не происходит. Если он находится в другой половине, он переписывается через накапливающий сумматор (прежнее содержимое которого сохраняется и устанавливается снова). Первоначальная запись

блока (на барабанах) теряется, т.к. содержащий ее сектор указывается в списке секторов, как "свободный".

2) Записать блок "b" на барабан. Если "b" уже на барабанах, ничего не происходит. Страница в ЗУ на сердечниках, содержащая эту информацию, теперь рассматривается как свободная.

3) Продублировать блок "b", назвать копию "b2". "b2" находится в ЗУ на сердечниках, если "b" был на барабанах и наоборот. Блок, раньше называвшийся "b2" (если он существует), теряется.

4) Очистить блок "b". Сектор или страница, занятая блоком "b", становится свободной.

5) Записать "n" блоков, начиная с "b", в "n" секторов, начиная с "s". Обмен продолжается в указанной зоне барабана, так что $n < 6$.

Блоки сохраняются в ЗУ на сердечниках, и первоначальная информация об этих секторах переписывается. Чтобы избежать потери времени на ожидание между последовательными передачами при передаче более чем одной страницы на барабан, необходимо, чтобы секторы, которых она касается, были все в одной группе дорожек барабана. Это приводит к двум ограничениям в этой и следующей командах

(I) $n \leq 6$ и

(II) секторы, записываемые или считываемые, все находятся в одной зоне.

Например, если $n = 4$, $b = 10$, $s = 3$.

Блок I0 в ЗУ на сердечниках переписывается в сектор 3

Блок I1 в ЗУ на сердечниках переписывается в сектор 4

Блок I2 в ЗУ на сердечниках переписывается в сектор 5

Блок I3 в ЗУ на сердечниках переписывается в сектор 0

(а не на зону I)

Если требуется передача в несколько страниц, для которой требуются 2 (или больше) группы дорожек, она должна быть осуществлена посредством двух (или более) команд.

6) Считать "n" блоков, начиная с сектора "s", и отметить новые записи этих блоков в блоках ЗУ на сердечниках, начиная с "b".

Обмен продолжается в указанной зоне барабана так, что $n \leq 6$. Содержимое секторов сохраняется, но все прежние блоки от "b" до $(b + n - 1)$ теряются.

4.4. ОСНОВНЫЕ ОБРАЩЕНИЯ К БАРАБАНАУ

Когда программисту нужен обмен с барабаном (указанная в команде экстракода или определенная неэквивалентность), передача начинается программой в фиксированном ЗУ, устанавливающей в "I" различные разряды в ЗУ "V". В следующих параграфах (описывающих по существу систему быстродействующих барабанов) все указания на определенные строки и разряды относятся к соответствующим строкам и разрядам в ЗУ "V" барабанов (см. раздел 13.4.).

Необходима следующая информация, чтобы указать ячейку на барабанах, с которой надо начать обмен.

(а) выбирается ли быстродействующий барабан (тип M.D.) или медленнодействующий барабан (барабан с таблицами) (строка 52, разряд 29),

(b) в какой стойке находится барабан (строка 52, разряды 28-26),

(c) какой требуется барабан внутри стойки (строка 48),

(d) какая требуется зона на барабане (строка 44),

(e) начальный регистр "θ" (строка 40, разряды 29-27).

Регистры "θ" (строки 0-31, разряды 29-27) дают угловое положение каждого из быстродействующих барабанов. Есть 6 секторов на каждой зоне для этих барабанов, и поэтому числа на регистрах устанавливаются только в пределах 0-5. Они устанавливаются машиной автоматически.

(f) сколько блоков информации нужно передать (строка 36).

Для передачи, более чем одного блока, все сектора должны быть в одной зоне. Если необходимо переменить выбранную зону (т.е. нужна другая группа дорожек, барабан или стойка) нужно время примерно в 1,5 мсек для "успокоения", прежде чем начнет выполняться следующий обмен. Во многих случаях (например, передача нескольких страниц или переписывание в следующий доступный свободный сектор) полезно иметь возможность узнать, возможно ли передавать в следующий сектор или с него, не ожидая полного оборота барабана. Разряды "2-1,5" (строки 0-31, разряд 26) предоставлены для этой цели. Соответствующий триггер сбрасывается на ноль, когда содержимое соответствующего регистра "θ" меняется и устанавливается на единицу, когда уже поздно выбирать следующий сектор для немедленной передачи. Чтобы дать программе постоянной памяти время завершить необходимую подготовку, для этого разряда выделено время на установку в 0,1 мсек, т.е. если в этом разряде прочтен 0, программа располагает по крайней мере 0,1 мсек, чтобы выбрать следующий сектор для немедленной передачи. Это время на установку 0,1 мсек нужно, конечно, рассматривать как максимальное возможное время, иначе разряд нужно проверять снова.

Чтобы указать страницу или страницы в ЗУ на сердечниках для обмена с барабанами, необходимо, чтобы регистр или регистры адреса страниц были установлены следующим образом:

разряд	22	21	20	19	18	17	16	15	14	13	12
	1	1	1	1	0	1	1	1	δ	δ	δ

Для передачи одного блока разряды 14-12 должны все содержать нули. Для передачи нескольких блоков (а также обмена с медленнодействующими барабанами) разряды 14-12 должны быть установлены в зависимости от страниц, на которые повлияет передача. Так, например, для передачи 2048 слов нужно установить разряды 14-12 любых четырех регистров адреса страниц соответственно на 000, 001, 010 и 011.

Разряд 23 - разряд блокировки, обычно установлен на единицу.

Когда требуется начальный сектор и число блоков указаны в строках 36-52, разряды управления в строке 56 должны быть установлены в "1" для того, чтобы начать пе-

редачу. Для этого необходимо:

- (a) установить пусковой разряд,
- (b) указать, нужна ли та же самая зона, что и последняя выбранная или другая и
- (c) указать, будет ли производится запись или считывание.

После завершения каждого обмена происходит автоматическое прерывание работы. Для передачи нескольких блоков периодически проверяются разряды "законченных блоков", и каждый раз, когда оказывается, что блок был использован при обмене, соответствующие регистры адреса, разряд блокировки и списки страниц и секторов приводятся в соответствие с новым положением. Автоматическое прерывание работы при передаче нескольких блоков производится, когда совершается обмен со всеми блоками, а не после передачи каждого блока.

После того как начат обмен, передача происходит автоматически, так что слово передается примерно каждые 4 мсек, а ЗУ на сердечниках бывает занято в каждом цикле примерно каждые 2 мсек из этого времени. Если есть еще какие-нибудь ссылки на ЗУ на сердечниках, обычные операции продолжают замедленным темпом, пока совершается передача, так что обмен с барабанами имеет преимущество.

Невозможность обмена с барабанами указывается тогда, когда передача не завершена в течение данного времени (около 25 мсек) после того, как указана в программе. Это может произойти, например, если указана несуществующая величина "θ" (например, θ = 6 или 7).

Если в команде указан обмен, использующий некоторую стойку, а эту стойку выбрать нельзя (это возможно лишь в результате ошибки, т.к. организующая программа может разрешить только передачу на существующие барабаны), происходит автоматическое прерывание работы и устанавливается в "1" разряд отсутствия стойки барабана. Если стойка выбрана правильно, но барабан внутри стойки выбран неправильно, вырабатывается сигнал невозможности обмена с барабаном. Происходит считывание единицы, если установлен разряд "стойка барабана отсутствует"; точно также необходимо записать единицу, чтобы снова восстановить разряд. (Аналогично для разрядов "передача на барабанах закончена" и "передача на барабанах невозможна"). На каждый из 24 разрядов информации, на барабане есть разряд четности. (Разряд четности таков, что общее число разрядов нечетно). Разряд четности обмена с барабаном (строка 0, разряд 30 ЗУ "v" Центральной вычислительной машины) устанавливается в "1" и происходит прерывание работы, если обнаруживается неправильная четность:

- (a) в содержимом страницы ЗУ на сердечниках, участвующей в записи при обмене с барабаном, или
- (b) в информации сектора барабана, предназначенном для считывания.

В любом случае, когда передача завершается, разряд четности для каждого слова проверяется, но не вырабатывается новый разряд четности в случае, если обнаруживается неправильная четность.

В ЗУ "v" не дается указаний на то, какое слово имеет неправильную четность.

Раздел 5 - ОПИСАНИЕ ВНЕШНИХ УСТРОЙСТВ

5.1. СТРУКТУРА ВНЕШНИХ УСТРОЙСТВ

5.1.1. ОРГАНИЗУЮЩАЯ ПРОГРАММА

Машина Атлас сконструирована так, что с ней может работать большой набор внешних устройств. Каждое такое устройство имеет специальное электронное оборудование, прямо связанное с ним для его управления, которое служит в качестве буферного ЗУ для обмена информацией.

Организация всех команд для внешних устройств производится программами постоянной памяти, которые называются организующими программами. Существуют отдельные организующие программы для барабана, магнитных лент, других внешних устройств и операций, выполняемых помимо работы основной машины; там, где необходимо, они образуют последовательности команд. Общая организующая программа согласует главные программы с этими организующими программами. Все команды внешнего оборудования проходят через ЗУ "V". Это может осуществляться или обменом информацией между ЗУ на сердечниках и ЗУ "V" (например, для устройств с бумажной лентой), или передачей слова в соответствующий регистр ЗУ "V", определяющий требуемую операцию (например, для устройств с магнитной лентой). Подробности даны в разделе I3.

Все устройства должны иметь кнопку "отключения" и соответствующую белую лампочку, которая загорается тогда, когда устройство отключено.

Имеются также соответствующие триггеры в ЗУ "V", указывающие состояние, в которое переведено каждое устройство. Общая организующая программа может считывать и устанавливать эти триггеры и таким образом включать и выключать эти механизмы и обнаруживать изменения, вызванные работой оператора.

Первая информация на каждой ленте ввода или пачке карт - это заголовок, в котором может стоять:

или ПРОГРАММА
или ДАННЫЕ
или ИСПРАВЛЕНИЕ
или СВОБОДНО (только для магнитных лент)
или КОПИЯ

Когда вводятся ленты или карты, относящиеся к свободным устройствам, которые включены, заголовки должны немедленно прочитываться и записываться, после чего дальнейшие действия машины зависят от типа используемого оборудования. Если на бумажной ленте или картах нет заголовка, оператор должен информировать организующую программу заранее; тогда эта лента или карты будут прочитываться под контролем программы. Пропуск заголовка является нежелательным.

Совсем необязательно использовать в данной программе какое-нибудь определенное устройство данного типа, нужно только, чтобы достаточное количество устройств этого типа было в наличии или во время выполнения программы, или (в случае устройств с медленной выдачей) позже. Эта выданная информация временно направляется на магнитную ленту, если нужно устройство в данный момент занято, а потом выдается на выходе. Общая организующая программа для каждой программы образует таблицу, которая содержит достаточную информацию для установления соотношения между механизмами, которые требуются для программы и которые имеются в наличии. Программа не начинает работать, пока нет достаточного числа свободных устройств, на что, если необходимо, дается указание на выходном устройстве оператора.

Процедура после того, как прочтен заголовок, такова:

1) Магнитные ленты

(а) Заголовок - ПРОГРАММА

В удобный момент (зависящий от очередности, загрузки вычислительной части, наличия свободных ячеек в главном ЗУ, и т.д.) программа считывается и выполняется. Заголовок должен определить число требуемых блоков главного ЗУ, а также привести в соответствие пометки, используемые для определения внешних устройств в тексте программы, с "внешними" пометками перфоленты или карт, или с определенным устройством, если оно почему-либо требуется.

Когда программа должна начать выполняться, делается проверка, прочтены ли уже требуемые заголовки. Если нет, то операторы должны получить команду поместить ленты или карты с данными "внешними" пометками в определенные свободные устройства.

(б) Заголовок - ДАННЫЕ

Если программа требует числовых данных с магнитной ленты, она обращается к заголовку, который ранее был отождествлен с внешним заголовком. Организующая программа приводит в соответствие требуемую ленту и механизм, в который эта лента введена.

(с) Заголовок - ИСПРАВЛЕНИЕ

Он относится к программе, которая находится в глав-

ном ЗУ и должно выполняться раньше других; она допускает вмешательство оператора.

Например, "Остановить программу ABC/2"
"Начать снова программу ABC/2"

(d) Заголовок - СВОБОДНО

Эти ленты могут быть использованы как рабочие ленты или ленты вывода. Оператор должен получить команду через выходное устройство оператора, ставить свободные ленты, когда это необходимо. Организующая программа проверяет, что введенная лента свободна, путем сверки ее заголовка и/или, сравнивая ленту со списком свободных лент, находящихся в машине.

Имеется экстракод, чтобы "освободить" магнитную ленту во время выполнения программы. Имеется также специальная программа, чтобы "освободить" магнитную ленту после ее удаления из вычислительного устройства, когда информация, записанная на ней, больше не нужна.

(e) Заголовок - КОПИЯ

Этот заголовок используется для любой короткой программы, которая просто "переписывает" из одного устройства в другое, например, с магнитной ленты в построчно-печатающее устройство. Такие программы могут выполняться и завершаться, пока выполняется другая длинная программа.

2) Перфоленты

После того, как заголовок был прочтен (для ленты ввода) и записан в ЗУ, лента считывается в главное ЗУ и преобразовывается при программном управлении. Если программа не требует этой ленты, она считывается в главное ЗУ или на магнитную ленту и позднее преобразуется при программном управлении. Лента с заголовком "ПРОГРАММА" переводится и выполняется немедленно, если это позволяют требования к памяти, очередность действий и т.д. Если используется нестандартная пробивка (например, 7-дорожечная лента с разрядом четности), это нужно указать в заголовке. Очевидные ошибки в ленте вызовут остановку, и оператор будет оповещен.

В добавление к вышеупомянутым категориям а, б, с, е заголовок может быть также таким:

(f) Заголовок - ОРГАНИЗАЦИЯ

Он используется оператором для изменения спецификаций программы (например, изменения требований к внешним устройствам), для установления или изменения очередности использования устройств и для любой другой "организации".

3) Клавишные устройства

Клавишные устройства используются:

- а) только, когда к ним обращаются, или
- б) когда им предшествует заголовок, отождествляющий программу. В этом случае считанная информация может снова быть записана во вспомогательное ЗУ.

4) Устройство считывания с карт

Процедура та же, что и для перфолент; заголовок указывает используемый при вводе код.

5) Устройство вывода

Общая организующая программа выбирает свободные устройства и сообщает об этом на выводное устройство оператора. Если устройства не являются свободными, требования на них временно направляются на магнитную ленту. Если это желательно, в заголовке программы может быть указано использование определенных устройств (например, если будут использоваться заранее напечатанные формы). Некоторые устройства можно использовать только кратковременно (например, ксерографическую печать); требования временно направляются на магнитную ленту и печатаются программой, имеющей заголовок КОПИЯ.

5.1.2. ВЫВОД ПО ТРЕБОВАНИЮ ОПЕРАТОРА

Это устройство, вероятно, будет на телетайпах, один для оператора магнитной ленты, другой для центрального пульта управления. Они будут использоваться всегда, когда операторам нужна информация, например такого рода:

- 1) причина остановки какого-либо устройства (нет бумаги, не сходится контрольная сумма на магнитной ленте, неправильная четность на перфоленте и т.д.),
- 2) причина, по которой прекращено выполнение программы (нет числовых данных, истек лимит времени и т.д.),
- 3) как используются магнитные ленты (пометить и отправить в ЗУ, лента нужна для печатания, лента "освобождается" и т.д.),
- 4) данные, которые должны быть выданы на определенные устройства ввода. Возможно также получить информацию путем прямого вмешательства оператора, с использованием тумблеров или лент с заголовком ОРГАНИЗАЦИЯ, например:
- 5) какая программа или программы выполняются в данный момент,
- 6) распределение внешних устройств (между программами) в данный момент,

7) список устройств, которые отключены и могут быть использованы.

Информация от инженерных тестовых программ также может выдаваться на центральном телетайпе. Требуется отдельное устройство вывода для ведения журнала работы машины (например, чтобы отмечать время, затраченное на каждую программу) и/или для учета.

5.1.3. ТИПЫ УСТРОЙСТВ

Предлагается, чтобы было предусмотрено присоединение к машине Атлас нижеследующих устройств. Дается также максимальное количество устройств любого типа, которые можно присоединить к машине. Возможно присоединить и другие устройства, кроме перечисленных ниже; их число не должно превышать 128.

(а) 32 устройства на магнитных лентах; из них до 8 могут работать одновременно,

(б) 32 магнитных барабана (тип M.D.5).

Они должны, как и медленнодействующие барабаны (барабаны с таблицами и программами), рассматриваться скорее как часть главного ЗУ, чем как внешние устройства. Главная память, а именно ЗУ на сердечниках, быстродействующие и медленнодействующие барабаны, ограничена общим пределом 220 слов,

(с) 2 медленнодействующих барабана (I.C.T.),

(д) 16 устройств для считывания с перфокарт, из них до четырех типа T.R.7 (1000 знаков в секунду) и двенадцать типа T.R.5 (300 знаков в секунду),

(е) 16 записывающих устройств,

(ф) 2 ксерографических печатающих устройства,

(г) до 16 устройств для вывода на перфокарты, из них до 4 быстродействующих (250 знаков в секунду) и 12 медленнодействующих (60 знаков в секунду),

(h) 2 печатающих устройства по 600 строк в минуту (I.C.T.),

(i) 2 часов,

(j) 2 графических выводных устройства,

(к) 4 устройства для считывания с перфокарт (600 карт в минуту),

(l) 2 устройства для вывода на перфокарты (100 карт в минуту),

(m) 16 5-канальных телетайпов (10 знаков в секунду типа "Крид-75").

5.1.4. СИГНАЛЬНЫЕ ТРИГГЕРЫ

С каждым внешним устройством связан "сигнальный" триггер в ЗУ"V". Цель этих триггеров - указать, что соответствующее внешнее устройство требует внимания. Установка на "I" одного из этих триггеров ведет к автоматическому прерыванию работы по программе. Один из этих триггеров устанавливается, например, когда прочтен знак с бумажной ленты и нужно знать, читать ли следующий знак

или остановить ленту. Если требуется прочитать следующий знак, уже прочтенный знак передается из триггеров информации - ЗУ"V" в главное ЗУ (через В-регистр или накапливающий сумматор) и сигнальный триггер сбрасывается на "0"; обе эти операции производятся соответствующей программой постоянного ЗУ. Затем прочитывается автоматически следующий знак и сигнальный триггер устанавливается снова на "I". Если новых знаков не требуется, программа устанавливается на "I" триггер снова. Аналогично соответствующий триггер ставится на "I", когда с магнитной ленты считывается адрес блока или маркер блока.

Сигнальные триггеры устройства для считывания с перфокарт соединены через схему "или", чтобы давать сигнал Р1 в разряде 30 строки 2 ЗУ"V" Центральной вычислительной машины. Таким образом, когда одно из устройств для чтения перфокарт требует внимания, устанавливается разряд Р1.

Когда на магнитной ленте прочитан адрес блока или маркер блока, устанавливается сигнальный триггер соответствующего канала. Схема "или" сигнальных триггеров каналов дает сигнал T для 28-го разряда 2-й строки ЗУ"V".

Остальные устройства разделяются на три "класса", соответственно тому, с которым из трех разрядов 2-ой строки ЗУ"V" Центральной вычислительной машины они связаны - Р2 (разряд 29), Р3 (разряд 27) или Р4 (разряд 26). Каждый из этих "классов" имеет связанную с ним строку в ЗУ"V" (соответственно строки 31, 30 и 29 "типа I4" ЗУ"V" внешних устройств), содержащие до 8 разрядов.

Устройства разделены на группы, каждая группа содержит некоторое число одинаковых устройств. Схема "или" сигнальных триггеров для каждой группы связана с разрядом в одной из этих трех строк. Другие цепи связывают разряды этих строк, также через цепи "или" с соответствующим разрядом (29, 27 или 26) строки 2 ЗУ"V" Центральной вычислительной машины.

Установка "I" разряда в строке 2 ЗУ"V" Центральной вычислительной машины вызывает автоматическое прерывание работы (процедура установления причины прерывания работы дана в разделе I4,3.2).

5.2. МАГНИТНЫЕ ЛЕНТЫ

5.2.1. УСТРОЙСТВО АМПЕКС

Схема ленты

Магнитофон Ампекс TM2 (FR 300) использует магнитную ленту шириной 1 дюйм (2,54 см). Лента имеет 16 дорожек, которые используются следующим образом:

12 дорожек информации

2 дорожки синхронизирующих импульсов

1 дорожка маркеров блоков, причем каждый маркер блока состоит из 12 разрядов

1 дорожка маркеров для ссылок; каждый маркер для

ссылки состоит из 5 разрядов (используются для целей адресации).

Маркеры для ссылок находятся примерно на 10 мм впереди соответствующих маркеров блоков. Эти маркеры помещены на двух внешних дорожках ленты.

Каждый синхронизирующий импульс связан с группой 6 разрядов информации: дорожки синхронизирующих импульсов помещены посередине шести соответствующих дорожек информации.

Информация записана на ленте в блоках информации или секциях по 512 слов. Каждой такой секции предшествует адрес, а за ней следует контрольная сумма. Этот адрес и контрольная сумма записываются на ленту в каналах информации. С каждым блоком информации связаны два маркера блоков - "ведущий" и "хвостовой". Адрес блока записывается "параллельно" ведущему маркеру блока для каждой секции ленты; "параллельно" каждому хвостовому маркеру блока на ленте находится нулевой адрес. Каждый адрес на ленте состоит из четырех полосок по 12 разрядов каждая; первые две полоски - нулевые, третья состоит из самого старшего разряда адреса и одиннадцати нулей, четвертая содержит младшие 12 разрядов адреса. Только тринадцать значащих разрядов считываются в регистр адреса блока. Ленты заранее специально размечаются в машине по адресам, прежде чем они будут использованы для работы; фиксированное положение адресов позволяет производить выборочную перезапись секций. Контрольные суммы имеют по 24 разряда с циклическим переносом и записываются в двух полосках; они используются для проверки правильности всех операций считывания и записи.

Каждая секция информации в 512 слов имеет в длину около 5,46 дюйма (139 мм) с промежутком в 2,3 дюйма (58,4 мм) между секциями. Расстояние между первыми разрядами ведущего и хвостового разрядов около 6,93 дюйма (176 мм), а маркеры блоков занимают около 0,032 дюйма (0,8 мм). Есть независимые головки записи и считывания, отстоящие приблизительно на 0,39 дюйма (10 мм); если лента не работает на полной скорости, она останавливается в положении, когда головки находятся в промежутке между секциями, готовая к записи или чтению следующей секции. Ленты имеют в длину около 3600 футов (1100 метров) и содержат около 5500 секций или два с половиной миллиона слов по 48 двоичных разрядов.

Выполнение

Нормальная скорость ленты около 120 дюймов в секунду (3 м в сек.); на каждой дорожке 375 двоичных разрядов на дюйм (около 15 на 1 мм). Это дает мгновенную скорость передачи в 90000 шестиразрядных знаков в секунду или одно 48-разрядное слово машины Атлас в 89 мксек (учитывая промежуток между блоками, эффективная скорость передачи около 64 тыс. знаков в секунду в каждом канале). Есть также быстрые операции перемотки ленты в прямом и обратном направлении при скорости около 180 дюймов в секунду (4,5 м в сек.).

Считывание возможно независимо от направления движения ленты, но запись возможна только при движении ленты вперед.

Управление

Машина Атлас может управлять максимально 32 устройствами на магнитной ленте. Каждое устройство соединено через один из восьми каналов, которые могут действовать независимо; каждый канал управляет одной операцией считывания, записи или поиска. Операции прямой и обратной перемотки ленты являются автономными и требуют канала только для того, чтобы начинать и, если нужно, заканчивать их.

Окончательная схема системы управления будет зависеть от конкретной установки. Для вычислительной машины Манчестерского университета с восемью магнитофонами и восемью каналами на каждое устройство любое устройство можно в любое время соединить с каждым каналом. Для установки с более чем восемью устройствами на магнитной ленте предполагается, что каждый канал будет управлять не более чем четырьмя механизмами, например, канал 0 управляет устройствами 0-3, канал 1 - устройствами 4-7 и т.д.

В ЗУ "V" магнитной ленты выделено достаточное число разрядов для управления любым из устройств, с каждого канала, но практические детали осуществления могут сделать это нежелательным. В остальной части этого описания предполагается, что каждый канал управляет не более чем четырьмя устройствами.

Обмен между ЗУ на сердечниках и лентой осуществляется через быстродействующее буферное устройство ЗУ на сердечниках, содержащее два слова на каждый канал. Для синхронизации ЗУ на сердечниках и этого буферного устройства введен быстродействующий регистр на триггерах. Одно слово можно передавать каждые 89 мксек и поэтому обращение в ЗУ на сердечниках может требоваться каждые 11 мксек, если работают все восемь каналов (в худшем случае каждые 8 мксек, если во всех передачах участвует один и тот же блок ЗУ на сердечниках). Из этого следует, что ЗУ на сердечниках занято в течение одного цикла 2 мксек, а в остальное время машина может производить обычные операции. Используется система очередности для сердечников, барабана и ленты, чтобы информация не терялась при обмене с барабаном или лентой вследствие того, что ЗУ на сердечниках может быть занято.

Защита

Бобина с лентой должна быть снабжена кольцом "Запись разрешена", прежде чем на эту ленту можно что-нибудь записывать. Ленты, содержащие постоянную информацию, не имеют такого кольца.

Переключатель тока записи также имеется на каждом устройстве; оператор может использовать его, чтобы отменить разрешение записи, даваемое кольцом. Записывать

можно только, если есть кольцо и включен ток записи. Ток записи автоматически выключается, если лента движется назад или при большой скорости перемотки в любом направлении. Он также автоматически выключается после завершения любого обмена с записью.

5.2.2. КОМАНДЫ ЭКСТРАКОДА ОБРАЩЕНИЯ К МАГНИТНЫМ ЛЕНТАМ

I. Идентификация лент

Если магнитная лента несет полезную информацию, описательный заголовок этой информации записан в секции 0 ленты. Перед использованием такой ленты программа или ее управляющая лента должны правильно указать название и приписать этой ленте номер V ($0 < V \leq 100$). Последующие команды ссылаются на ленту V этой программы: нормально V пишется в разрядах V_n команды. Номера лент больше 100 резервируются для особых случаев использования подпрограммами.

Слово "СВОБОДНО", употребляемое как заголовок, указывает, что данная лента не несет полезной информации и доступна для общего применения.

2. Обмен фиксированными массивами

Основные операции с лентой производят обмен информацией между секциями ленты и блоками информации по 512 слов главного ЗУ. Команды для работы с лентой, которыми пользуется программист, являются экстракодами, основанными на этих операциях. Наибольшая эффективность получается, если рассматривать ЗУ разделенным на блоки по 512 слов и использовать команды для обмена блоков с магнитной лентой.

Во всех следующих командах номера секций ленты, когда они указаны, имеют вид полного адреса.

2.1. Команды поиска и обмена блоков

В следующих командах параметр K ($0 \leq K \leq 6$) определяется последним восьмеричным разрядом адреса. Один блок информации передается или лента сдвигается на одну секцию, если $K = 0$ или 1.

- 1) Найти секцию s на ленте V и остановиться перед ней.
- 2) Продвинуть ленту V вперед на K секций.
- 3) Продвинуть ленту V назад на K секций.
- 4) Считать следующие K секций с ленты V на страницы ЗУ $P, P+1, \dots, P+K-1$.
- 5) Считать предыдущие K секций с ленты V на страницы ЗУ $P+K-1, P+K-2, \dots, P$.

6) Записать страницы ЗУ, $P, P+1, \dots, P+K-1$ на следующие K секций ленты V .

2.2. Организационные команды

7) Установка

Приписать номер V ленте, заголовок которой записан в ячейках $s, s+1$ и т.д. Если эта лента еще не готова, указать оператору, чтобы он ее ввел.

8) Ввод свободной ленты

Выбрать свободную ленту, записать на ней заголовок, записанный в ячейках $s, s+1$ и т.д., и приписать ей номер ленты V этой программы. (Операции 7 и 8 также можно осуществить путем команды на управляющей ленте или карте, которые могут образовывать часть данных).

9) Записать заголовок

Записать на ленте V заголовок, записанный в $s, s+1$ и т.д.

10) Прочитать заголовок

Считать заголовок ленты V в ячейки $s, s+1$ и т.д.

11) Снять ленту

Перемотать, выключить механизм ленты и указать оператору, чтобы он снял ленту.

12) Свободная лента

Стереть заголовок на ленте V и вернуть ленту к подчинению организующей программы для обычного использования. (Ленты можно также "освободить" с помощью управляющей ленты).

13) Высвободить ленту для другой программы

С ленты V стираются адреса данной программы и она становится пригодной для другой программы, не "освобождаясь" и не отключаясь.

Если $n \neq 0$ в командах 11-13 количество устройств с лентой, выделенных для программы, уменьшается на единицу.

14) Высвобождение устройств

Уменьшить на s количество устройств с лентой, зарезервированных для использования программой. Только отключенные механизмы можно высвободить от программы.

15) Изменение названия

Придать название s ленте, которая раньше называлась в этой программе лентой bw .

16) Определение объема информации

hw' - количество секций по 512 слов, доступных на ленте V (кроме секции 0). В полном слове адреса занимает полслова.

17) Местонахождение (см. также раздел 4.2)

После команд обмена блоков $s' = A/Q$, где A - адрес следующей секции ленты V . В полном слове адреса положение первого полслова $Q = 0$.

3. Обмены информацией переменного объема

3.1. Вид информации на ленте

Некоторые программы легче писать, если информацию на магнитной ленте можно обрабатывать в кусках разной дли-

ны. Для этой цели есть набор экстракодов, использующих блоки ЗУ на сердечниках как буферные устройства для обеспечения обмена информацией переменной длины.

Информация хранится на ленте в группах или цепочках слов; имеется маркер в 24 разряда для обозначения конца цепочки. При каждом обмене с записью на ленту записывается одна цепочка слов. При обмене со считыванием либо считывается определенное количество слов информации или происходит считывание до конца группы: в обоих случаях маркеры при обмене опускаются.

Часто несколько передач с записью образуют большую единицу информации, такую как "документ" или целая картотека, и может быть, желательно как-то это обозначить. Для этого имеются маркеры 7 порядков: от I до 7. Обычные цепочки могут заканчиваться маркером порядка I, записи, состоящие из нескольких цепочек, — маркером порядка 2, группы записей — маркером порядка 3 и т.д. Маркер порядка I в начале (или в конце) цепочки заменяется соответствующим маркером более высокого порядка, если эта цепочка — первая (или последняя) в документе или группе документов. Обмены со считыванием происходят до маркера указанного или более высокого порядка.

Два маркера, указывающие конец одной цепочки и начало следующей, соединяются на ленте в одно слово. Во время обмена с записью порядок маркера записывается как в маркере, следующем за только что записанной цепочкой, так и в маркере, предшествующем следующей цепочке. Для этой цели предназначены три разряда каждого маркера.

Остающиеся двадцать один разряд каждого маркера указывают соответственно длину предыдущей и следующей цепочки.

3.2. Вид команд

Работа с лентой переменной длины должна всегда начинаться пусковой командой. Она выбирает магнитную ленту, с которой можно работать, и тип операции — запись/чтение с движением вперед или чтение назад. Она также устанавливает буферное устройство ЗУ на сердечниках и обеспечивает при чтении, чтобы оно использовалось, когда в нем меньше S слов, причем S определяется оператором. Если возможно, S следует выбирать так, чтобы следующую секцию можно было считывать с ленты раньше, чем программа окончила работу с остающимися S словами. Если это можно сделать, программа не должна ожидать магнитной ленты: при обмене со считыванием должна быть только перепись из буферного устройства ЗУ на сердечниках в указанный адрес.

Если программа читает информацию большими группами, обработка которых отнимает много времени, то содержание в течение этого времени большого буферного устройства может оказаться расточительным. В этом случае пусковая команда может задать небольшой буфер. За некоторое время до того, как потребуются информация, можно дать команду выборки, требующую, чтобы буфер был заполнен, по

крайней мере, S словами, готовыми для предстоящего обмена со считыванием. Имеется отдельная команда обмена, работающая с лентой, которая была выбрана последней командой пуска или выборки. Она передает информацию из буферного устройства ЗУ на сердечниках в определенный программой адрес или обратно заранее выбранным образом: читать с движением вперед, читать назад или записывать. Чтобы изменить способ обмена, необходимо выполнить другую пусковую команду. Нужна отдельная пусковая команда для каждой ленты, для которой требуется обмен с переменным объемом информации, но затем команду выборки можно использовать для выбора номера ленты.

Пусковые операции всегда начинают обмен переменной длины со следующего слова на ленте или предыдущего слова в случае считывания с движением назад. Чтобы начать работу с определенным словом на ленте, операции пуска должна предшествовать команда поиска; в случае обмена со считыванием, выбранный адрес должен быть адресом маркера в конце цепочки. Используя команду пуска для перехода от одного вида обмена к другому, не обязательно останавливаться на маркере.

Операции записи с лентой переменной длины не обеспечивают средств перезаписи избранных слов в секции магнитной ленты: вместо этого полностью формируется новая секция в буферном устройстве ЗУ на сердечниках, а прежнее содержимое ленты не сохраняется. Если требуется сохранить начало первой секции, это можно сделать, заменив пусковую команду записи пусковой командой считывания с движением вперед при $S = 0$.

Имеется специальная команда пуска для считывания информации, записанной при обмене блоков без маркеров в конце цепочек.

4. Команды переменной длины информации

4.1. Команды пуска

18) Начать считывание вперед

Выбрать ленту В для считывания с движением ленты вперед с использованием обменов с переменным объемом информации, начиная со следующего слова на ленте. В дальнейшем обеспечить, чтобы по крайней мере S слов находилось в буферном устройстве ЗУ на сердечниках в ожидании обмена ($0 \leq S \leq 4096$).

19) Начать считывание назад

Выбрать ленту В для считывания с движением ленты назад с использованием обменов с переменным объемом информации, начиная со следующего слова на ленте. В дальнейшем обеспечить, чтобы по крайней мере S слов находилось в буферном устройстве ЗУ на сердечниках в ожидании обмена ($0 \leq S \leq 4096$).

20) Начать запись вперед

Выбрать ленту В для записи с движением ленты вперед с использованием обменов с переменным объемом информации, начиная со следующего слова на ленте. Буферные блоки ис-

пользуются по мере надобности. Маркер К записывается перед первым словом информации ($1 \leq K \leq 7$).

21) Выбрать

Выбрать ленту В для последующих операций с переменным объемом информации тем способом, который ранее определен для этой ленты. Если S - не нуль в команде выборки в операции считывания, то команды магнитной ленты начинаются так, как требуется для того, чтобы по крайней мере S слов были доступны в буферном устройстве ЗУ на сердечниках ($0 \leq S \leq 4096$).

22) Начать чтение вперед от фиксированных блоков

23) Начать чтение назад от фиксированных блоков

Команды 22 и 23 те же, что и 18 и 19, за исключением того, что они начинают чтение переменного объема информации для лент, которые записаны при обмене блоками, и поэтому не разделены на цепочки.

4.2. Команды обмена

24) Обмен

Обмен bw слов между адресами ЗУ, начинающимися на S, и избранной лентой, тем способом (считывание с движением вперед, чтение назад или запись), который соответствует этой ленте. Обмен кончается маркером bk.

При записи bw слов из ячеек S, S+1, ..., S+bw-1 записываются в следующие bw ячеек на избранной ленте. Маркер bk записан на ленте после них.

При считывании обмен продолжается, пока bw слов информации не будут считаны или пока не встретится маркер $\geq bk$, в зависимости от того, что произойдет раньше. bw' = количеству, фактически, считанных слов информации. bk' = 0, если не был встречен маркер $\geq bk$. bk' = m, если маркер m ($\geq bk$) сканчивал обмен или шел непосредственно после слова bw.

При считывании с движением вперед следующие bw' слов считываются с ленты в ячейки ЗУ S, S+1, ..., S+bw'-1.

При считывании назад предшествующие bw' слов прочитываются с ленты в ячейки ЗУ

$$S + bw - 1, S + bw - 2, \dots, S + bw - bw'.$$

Если b = 0 при считывании обмен продолжается, пока не встретится первый маркер, как если бы bw было равно bw'. При чтении назад это означает, что bw' слов прочитываются в ячейки ЗУ S + bw' - 1, S + bw' - 2, ..., S.

25) Пропуск

Пропустить bw слов, кончающихся маркером bk.

Пропуск выполняется так же, как обмен, за исключением того, что слова не передаются. При записи адреса bw на ленте пропускаются и после них записывается маркер bk. Однако важно, что прежнее содержимое ячеек с этими адресами, будь то информация или маркеры, не сохраняется на ленте, кроме тех случаев, когда пропускаются целые секции ленты по 512 слов.

При записи пропуск продолжается пока не пропущены bw слов информации или пока не встречен маркер $\geq bk$, смотря по тому, что случится раньше.

bw' = числу фактически пропущенных слов информации

bk' = 0, если не встречен маркер $\geq bk$

bk' = m, если маркер $\geq bk$ сканчивал обмен или шел непосредственно после слова bw.

Заметим, что пропуск значительно менее эффективен, чем поиск в установлении положения ленты и должен поэтому использоваться лишь для немногих секций на ленте.

26) Пометить

Возможно только при записи.

Записывает маркер К после последнего слова на выбранной ленте. Этот маркер заменяет любой маркер, который ранее находился на ленте в этом месте.

После записи цепочки на ленте иногда обнаруживается, что достигнут конец группы. Команда "пометка" может быть затем использована для изменения маркера в конце цепочки. Ее можно использовать снова, если позднее обнаруживается, что достигнут конец группы еще более высокого порядка.

27) Остановка

Остановить операции с переменным объемом информации на ленте В.

После того, как операции с переменным объемом для любой данной ленты прекращены, может быть дана команда остановки. Она высвободит буферные блоки, связанные с этими операциями. После операций записи она заставит немедленно переписать последнюю секцию из буферного устройства на магнитную ленту, но это только можно сделать любой из следующих операций: начало, поиск, вывод, высвобождение ленты, обмен блока.

17) Местонахождение (см. также раздел 2.2(17))

После обмена с переменным объемом $s' = A/Q$, где Q - адрес внутри секции текущего маркера, на котором остановилась лента на ленте В, или, если она остановилась не на маркере, адрес следующего слова при движении вперед. A' - адрес секции, содержащей слова Q.

28) Поиск слова

Поиск слова Q, секция А ленты В, где $s = A/Q$.

29) Запись индекса

Записать s в ячейку В индекса для избранной ленты $0 \leq B \leq 7$. Нормально s будет адресом на ленте, найденным командой местонахождения. Когда лента выводится, индекс записывается в секции 0.

30) Прочитать индекс

s' - содержимое ячейки В индекса избранной ленты $0 \leq B \leq 7$.

Нормально за этим следует команда поиска слова.

Заметьте, что команды касающиеся индекса, можно использовать при работе с фиксированным блоком, при условии, что им предшествует команда выборки для определения номера ленты.

Заметьте также, что команды индекса ограничены определенной лентой и их нельзя использовать для ссылок на ленту, являющуюся продолжением данной.

При использовании обменов переменных объемов информации используется специальный маркер для обозначения конца ленты и даются команды для перемены ленты.

5.2.3. ОЧЕРЕДНОСТЬ ОБРАЩЕНИЯ К МАГНИТНЫМ ЛЕНТАМ

Команды для магнитной ленты разбиваются на некоторое число полусловных команд следующего вида:

- (1) найти блок n ленты в устройстве m ,
- (2) считать вперед блок n ленты в устройстве m на страницу p главного ЗУ,
- (3) считать назад блок $(n-1)$ ленты в устройстве m на страницу p главного ЗУ,
- (4) записать страницу p главного ЗУ в блок n ленты в устройстве m ,
- (5) перемотать ленту в устройстве m ,
- (6) продвинуть вперед ленту в устройстве m .

Если одна из этих команд не может быть выполнена немедленно, она помещается в "список очередности". Прежде чем выполнять команду, необходимо удостовериться, что:

- (1) указанное устройство не требуется командой, стоящей в списке очередности выше; (2) указанный блок информации в ЗУ на сердечниках не заблокирован.

Команды, относящиеся к одному и тому же магнитофону, выполняются раньше, чтобы ленте не приходилось останавливаться между командами, пока не требуется изменить направление движения.

Максимальная вместимость списка очередности ленты - 64 команды. После того как выполнится каждая команда, команды, стоящие в очереди перед ней, сдвигаются на одну ячейку. Если список очередности заполнен и встречается новая команда для ленты, программа задерживается, пока не будет выполнена одна из команд в списке очередности.

5.2.4. ПРОГРАММА ПОИСКА АДРЕСОВ БЛОКОВ

Программа поиска адреса блока - это программа прерывания управления. Она поддерживает в памяти список текущих адресов всех устройств с магнитной лентой, соединенных с вычислительной машиной. Когда дана команда поиска для адреса блока в устройстве, лента включается на канал и движется в соответствующем направлении при скорости, зависящей от расстояния, которое она должна пройти. Таким образом, если есть только несколько блоков между требуемым адресом и текущим адресом, то используется нормальная скорость в 120 дюймов в секунду, но там, где нужно, используется скорость перемотки ленты в ту и другую сторону, равная 180 дюймов/сек.

При приближении к указанному блоку скорость уменьшается до нормальной, если ранее использовалась более высокая скорость. Через несколько секунд каждый адрес блока в отдельности сравнивается, пока не будет найден искомым. Для этой цели введен регистр адреса блока. Каждый раз, как встречается новый маркер блока, адрес блока автоматически считывается в этот регистр, происхо-

дит прерывание и устанавливается триггер прерывания маркера блока. За одну мсек адрес блока должен быть проверен и сравнен, и дано указание, найден ли искомым блок или нужен дальнейший поиск.

Последовательный контроль программой постоянного ЗУ введен для того, чтобы удостовериться, что каждый адрес блока таков, как ожидалось, т.е. ведущий адрес блока на единицу больше или меньше, чем прочтенный перед ним ведущий адрес, смотря по тому, движется ли лента вперед или назад, и что следующий (или нулевой) адрес блока прочитывается после каждого ведущего адреса блока.

Прерывание маркера блока автоматически запрещается во время быстрой перемотки ленты. В течение около пяти секунд, которые должны быть выделены на переход от быстрой скорости к нормальной, прерывание маркера блока происходит, но вследствие возможности того, что адреса были прочтены неправильно, программа фиксированного ЗУ их не проверяет.

Если программа поиска не обнаруживает нужного адреса блока после трех попыток, т.е. если то место, где должен быть блок, трижды пройдено читающей головкой, программе управления указывается на наличие состояния ошибки.

5.2.5. ОСНОВНЫЕ ОПЕРАЦИИ С МАГНИТНЫМИ ЛЕНТАМИ

Все команды с магнитной лентой, указанные программой, - это экстракоды (раздел 5.2.2.), которые разбиваются на полусловные команды и помещаются в список очередности (раздел 5.2.3.). Выполнение основной операции с магнитной лентой начинается записью соответствующих разрядов в ЗУ "V" магнитной ленты. Ссылки в следующих параграфах на определенные строки и разряды относятся к соответствующим строкам и разрядам в ЗУ "V" магнитной ленты.

Есть восемь независимых каналов с номерами от 0 до 7, каждый из которых соединяет машину с одной из групп по 4 из 32 устройств с лентой. С помощью этих каналов возможно выполнять одновременно до восьми команд поиска, чтения или записи для магнитной ленты, а также команды перемотки вперед или назад на остающихся устройствах. Каждое устройство снабжено красной лампой, которая зажигается, когда устройство выключено. Есть также соответствующий разряд в ЗУ "V" (строки 24 и 23, разряды 24-39) для каждого устройства, которое установлено на единицу, когда устройство выключено. Красная лампа гаснет и триггер сбрасывается в ноль, чтобы указать "включено", когда лента правильно введена, дверь устройства закрыта и нажата кнопка "включено". Триггер ставится в состояние "устройство выключено" (и красная лампа автоматически зажигается, когда устройство больше не нужно). Невозможно для программиста дать команду "включено".

С каждым каналом в ЗУ "V" связаны следующие регистры и триггеры:

Регистр адреса текущего блока (P.V.A.R.)

Регистр управления лентой (T.C.R.)

Регистр выборки магнитофона

Триггер прерывания адреса блока

Триггер прерывания магнитофона.

Управление магнитной лентой осуществляется записью и считыванием с этих регистров.

Устройство с магнитной лентой соединяется с одним из восьми каналов путем указания того, какое из четырех устройств, обслуживаемых этим каналом, требуется, и записи соответствующих разрядов в соответствующий регистр выбора магнитофона (строки 16-23, разряды 25 и 24. Разряды 28-26 нормально вводятся, как разряды соответствующего канала). Соединение устройства с каналом автоматически объединяет от этого канала всякое другое устройство, соединенное с ним в этот момент. Невозможно считывать из этого регистра, поэтому программа фиксированного ЗУ должна хранить список, указывающий какое устройство соединено с каждым каналом. Устройство с лентой управляется тогда вычислительной машиной через регистр управления лентой для этого канала.

Для целей поиска определенного адреса блока на ленте служат регистры адреса текущего блока (строки 0-7, разряды 36-24 P.V.A.R.). Каждая строка содержит тринадцатиразрядный адрес блока для своего, соответствующего ей канала. Когда на ленте встречается маркер блока, адрес блока (который равен нулю для следующего маркера блока) считывается в P.V.A.R. триггер прерывания адреса блока устанавливается для соответствующего канала (строка 26, разряды 31-24), и происходит автоматическое прерывание работы. "Прерывание вследствие остановки" происходит, когда движение ленты остановлено.

Данные регистра P.V.A.R. могут считываться машиной и вдобавок, для целей адресации и переадресации одних только лент, машина может записывать в P.V.A.R. в канал 7. Триггеры прерывания адреса блока устанавливаются в "I" координатором ленты; они сбрасываются в "0" путем записи единицы из машины.

Регистры управления лентой (строки 8-15) получают команды от машины и содержат информацию, касающуюся управления лентой по соответствующим каналам, следующим образом:

(а) Разряд "несовпадения контрольной суммы" (разряд 24) устанавливается на I координатором ленты; если обнаруживается несовпадение контрольной суммы (раздел 5.2.6.); он может считываться машиной и сбрасываться в "0" путем записи I из машины.

(б) Разряд "обмена коротких блоков" (разряд 25) устанавливается в "I" и сбрасывается в "0" координатором ленты и машина может только считывать его. Он используется, когда лента типа Орион считывается на машину Атлас (Раздел 5.2.8.).

(в) "Конец ленты" (разряд 26) появляется, когда устройство воспринимает покрытую металлом наклейку в лю-

бом конце ленты. Этот разряд машина может только считывать: если получен сигнал конца ленты, когда лента движется с нормальной скоростью, то происходит прерывание работы "по причине неисправности магнитофона".

(г) "Запись разрешена" (разряд 27) указывает, что кольцо "разрешения записи" находится на катушке ленты и переключатель тока записи включен для устройства, соединенного с каналом. Этот разряд машина может только считывать; он равен единице лишь, когда удовлетворяются оба условия.

(д) "Конец обмена" (разряд 28). Этот разряд устанавливается, чтобы окончить обмен, если обнаружена какая-либо ошибка. Это возможно (I), если недостаточное число страниц отведено для обмена с лентой типа Орион (II), если получено несовпадение между адресом обмена и регистрами адреса страниц, или (III), например, в обмене со считыванием вперед, ведущий адрес проверяется и оказывается неверным. Если движение ленты при ошибке не прекращено, можно дать сигнал "конца обмена", чтобы координатор ленты не отнимал больше время у координатора сердечников. Можно также применять его для сбрасывания в "0" триггера "записи" и отключения тока записи, если автоматическое отключение тока записи в конце обмена с записью почему-либо не произошло.

(е) "Запись" (разряд 29). Разряд записи должен быть установлен, прежде чем имеет место обмен с записью. Он автоматически сбрасывается в "0", когда прочтен следующий маркер. Для предохранения от переполнения лент, вызванного ошибкой в этом регистре, программа фиксированного ЗУ проверяет разряд записи, чтобы убедиться, что он правильно сбросился, когда завершен обмен.

(ж) "Окончить считывание перед следующим адресом блока" и "Считывать со следующего адреса блока" (разряды 30 и 31). Разряд "Считывать со следующего адреса блока" устанавливается для обмена со считыванием, когда читающая головка находится в междублочном промежутке перед нужным блоком. Когда происходит прерывание адреса блока, программа фиксированного ЗУ устанавливает разряд "Окончить считывание перед следующим адресом блока". Тогда обмен со считыванием заканчивается в конце блока.

(з) "Нормальное считывание" и "восстановительное считывание". Разряд "нормального считывания" устанавливает порог чувствительности так, чтобы сигналы ленты слабее определенной величины не воспринимались и не считывались ложные сигналы. После несхождения контрольной суммы или, если при выполнении обмена со считыванием не удалось прочесть все 512 слов, во время выполнения обмена системой управления "восстановительное чтение" может дать команду производить дальше обмен со считыванием при пониженном пороге чувствительности для восстановления частично записанных сигналов. Если этот обмен успешно закончен, адрес этого неудовлетворительно сработавшего блока ленты запоминается.

(и) "Остановка" (разряд 36). Через I,5 мсек после считывания адреса блока рассматривается разряд остановки и, если он установлен в "I", останавливает ленту. Но-

рмально, когда нужно остановить ленту, этот разряд устанавливается в "I" программой прерывания адреса блока фиксированного ЗУ в течение 1,5 мсек от чтения адреса блока. После этого промежутка времени никакие дальнейшие изменения в регистре управления лентой невозможны, пока лента не остановится и не произойдет прерывание вследствие остановки. Это прерывание использует тот же сигнальный триггер, что и прерывание адреса блока, но отличается от него тем, что содержимое Р.В.А.Р. остается неизменным со времени предыдущего прерывания. Лента должна быть остановлена прежде чем может начаться выполнение команды, изменяющей направление движения ленты.

(к) "Не начинать" и "Обратное движение" (разряды 36 и 37),
 (л) "Вперед" и "Обратное движение" (разряды 38 и 39),
 (м) "Нормальная скорость" и "Большая скорость" (разряды 40 и 41).

Эти последние три регистра устанавливаются, чтобы указать нужное направление движения и скорость (120 или 180 дюймов/сек) ленты.

Ток записи автоматически прекращается при установке разряда обратного движения или большой скорости.

Триггер прерывания магнитофона (строка 27, разряды 31 и 24) устанавливается в "I" и происходит прерывание, если произошла ошибка и она автоматически обнаружена на одном из магнитофонов, включенных в канал. Этой ошибкой может быть перегрузка реле, предупреждение о петле ленты, достижение конца ленты при нормальной скорости, отдача неправильных команд и т.д. Программа управления ленты будет пытаться очистить разряд ошибки, записывая в него единицу, но если разряд остается установленным, печатается указание на то, что в определенном устройстве произошла ошибка. Если разряд ошибки можно сбросить в "0", т.е. если ошибка исправилась сама себя, устройство может выполнять дальнейшие команды.

Когда программист закончил работу с бобиной ленты, соответствующее устройство устанавливается на быструю перемотку в прямом или обратном направлении и пуск. Затем оно отключается и канал становится свободен для другого механизма. Перемотка продолжается после снятия устройства с канала, пока не будет достигнута металлическая наклейка на конце ленты, затем на устройстве загорается красная лампа выключения.

Для указания страницы, используемой при обмене с лентой, один из регистров адреса страницы должен быть установлен так:

разряд	22	21	20	19	18	17	16	15	14	13	12
	I	I	I	I	I	δ	δ	δ	ε	ε	ε

Разряды 17-15 указывают соответствующий канал ленты. В разрядах 14-12 установлены 000 для обмена с движением вперед и III для обмена со считыванием назад (при считывании ленты типа Орион с блоками более 512 слов в них устанавливаются 000, 001, и т.д., или III, II0 и

т.д.) для разных страниц, затрагиваемых обменом (раздел 5.2.8).

Разряд 23 используется как разряд блокировки и нормально установлен в "I". Когда обмен закончен, разряд блокировки должен быть очищен и Р.А.Р. сбрасывается в "0", чтобы дать правильный адрес главного ЗУ для блока информации.

Для инженерных целей вырабатывается адрес обмена с магнитной лентой и хранится в машине в следующем виде:

разряд	22	21	20	19	18	17	16	15	14	13	12	11	10	9
	I	I	I	I	I	δ	δ	δ	ε	ε	ε	α	α	α
	8	7	6	5	4	3								

α α α α α

Разряды 17-15 и 14-12 имеют те же значения, что и выше. Разряды 11-3 используются, как адреса, внутри страницы передаваемого слова, а так же, как счетчик со значениями до 511.

Для обмена при движении направления вперед они первоначально все установлены в "0", а для обмена в обратном направлении они все первоначально установлены в "I".

Разряды 2-0 всегда содержат "0" и вообще не нужны. Операция "совпадения" между адресом этого обмена и регистрами адреса страниц проводится только для разрядов 22-12. Если происходит прерывание вследствие "несовпадения", соответствующий адрес обмена записывается в ЗУ "V" на сердечниках и разряд "управления барабаном, лентой или прерыванием" устанавливается в единицу, чтобы установить причину прерывания.

Если обнаружена неправильная четность слова на странице ЗУ на сердечниках, разряд четности магнитной ленты (строка 0, разряд 29 ЗУ "V" центрального вычислительного устройства) устанавливается в "I" и вызывает прерывание работы. Обмен закончен. В ЗУ "V" не дается никаких указаний о том, какое слово внутри страницы имеет неправильную четность.

5.2.6. КОНТРОЛЬНОЕ СУММИРОВАНИЕ ПРИ ОБРАЩЕНИИ К МАГНИТНЫМ ЛЕНТАМ

Обмен информацией между ЗУ на сердечниках и буферным устройством осуществляется по одному 48-разрядному слову за раз, пока не передан блок в 512 слов. В течение этого процесса полуслова по 24 разряда складываются с "циклическим переносом", пока не получится проверочная сумма для всего блока информации.

При операции записи контрольная сумма записывается немедленно после блока. Во время операции записи читаю-

шая головка, которая "следует" за пишущей головкой, считывает обратно информацию и вычисляет другую контрольную сумму. Сумма, записанная на ленте, также считывается обратно. Когда операция записи закончена, сравниваются три контрольных суммы, а именно, вычисленная во время обмена с записью, вычисленная во время обратного считывания и, наконец, записанная на ленте и считанная затем обратно. Если есть расхождение, разряд "неправильной контрольной суммы" устанавливается в "1" в соответствующем регистре управления лентой.

При операции считывания контрольная сумма вычисляется по мере того, как прочитываются данные, и она сравнивается с контрольной суммой, считанной с ленты. Расхождение вызывает установление разряда неправильной контрольной суммы в соответствующем регистре управления лентой. Программа фиксированного ЗУ делает еще две попытки считать этот блок; во второй из них используется команда "восстановительное чтение", при которой повышается коэффициент усиления. Если обе эти попытки не имеют успеха, об этом уведомляется программа управления.

Если во время операции считывания или записи обнаруживается неисправный блок, соответствующий номер блока запоминается и позднее записывается на каком-либо устройстве вывода (например, перфкарте) для использования в дальнейшем в программе переадресации.

5.2.7. РЕГИСТРЫ АДРЕСОВ БЛОКОВ МАГНИТНЫХ ЛЕНТ

Блоки информации на магнитных лентах имеют 512 слов. Каждая передача информации с использованием данного регистра управления лентой всегда происходит в блоки или из блоков ЗУ на сердечниках с фиксированными адресами блоков. Каждый регистр управления лентой имеет связанный с ним регистр, который содержит адрес этого блока. Дальнейшие разряды этих регистров используются для подсчета числа слов, передаваемых в течение операции. Эти регистры эффективно содержат следующие разряды:

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4
0	1	1	1	1	1	1	1	δ	δ	δ	ε	ε	ε	ε	ε	ε	ε	ε	ε
3	2	1	0																
ε	0	0	0																

где разряды от 12 до 23 относятся к адресу блока в ЗУ на сердечниках, разряды от 13 до 23 фиксированы и поэтому могут вырабатываться, необязательно существуя в этом регистре. Разряды 13-15 указывают регистр управления лентой. Разряды 3-12 используются для технических

целей только при обмене. Разряды 0-2 необязательны. При обмене необходимо для регистра адреса страницы, чтобы на нем был установлен адрес блока, соответствующий регистру управления лентой. Это делается программой управления магнитной лентой и после того, как обмен закончился, регистру адреса страницы дается тот адрес блока главного ЗУ, который он имел раньше. Во время обмена ставится в "1" разряд блокировки соответствующего регистра адреса страницы.

5.2.8. АДРЕСАЦИЯ И ПЕРЕАДРЕСАЦИЯ МАГНИТНЫХ ЛЕНТ

Машина Атлас нормально использует заранее проверенные и размеченные по адресам ленты с фиксированной длиной блока в 512 слов. Предварительная проверка и разметка нужны для того, чтобы ленты были надежными и все неисправные места на ленте были выведены из употребления.

Атлас также проверяет и размечает по адресам свои ленты. Для этой цели имеется регистр управления адресами ленты (Т.А.С.Р.) в ЗУ "V" магнитной ленты (строка 28). Этот регистр, существующий только в канале 7, дает команды, нужные для разметки ленты, так же, как и запись отметок для ссылок, запись адресов блока, проверка адресов блока и проверка блоков на ленте путем записи на нее единиц.

Две программы управления прерыванием в главном ЗУ соединены с Т.А.С.Р., а именно, программа разметки по адресам и программа переадресации. Программа разметки по адресам проверяет ленту на неисправные места, записывает исправные блоки с их отметками, и последовательно нумерует адреса блоков. Программа переадресации, которая используется, когда неисправные блоки обнаружены при нормальной работе, устраняет эти блоки с ленты и последовательно меняет номера остающихся адресов блоков.

Для разметки или переадресации ленты машине дается команда с помощью устройства ввода. Тогда программа управления печатает команду оператору подготовить магнитофон "К" для разметки по адресам и переадресации и установить на него катушку ленты. После того, как оператор это сделал, программа управления запрашивает триггер подготовки устройства в Т.А.С.Р., установлено ли оно для адресации; если да, то машина начинает работать с программой разметки или переадресации.

При выполнении программы разметки по адресам проходит примерно шесть футов ленты, прежде чем записывается первый маркер для ссылок. Маркер для ссылок записывается на ленте, когда в разряд 31 Т.А.С.Р. записывается единица.

В то же время, когда начинается запись первого ведущего маркера для ссылок, устанавливаются разряды счетчика разрешения, адресов ленты и записи единиц. Затем два просчета выполняются машиной автоматически. Первый указывает, когда записывать следующий маркер для ссылок, а

второй, — когда записывать ведущий маркер для следующего блока. Эти просчеты автоматически заканчиваются, когда записывается ведущий маркер для ссылок. Программа формирует следующий адрес (ноль для следующего адреса) для записи и отправляет его в Р.В.А.Р. для канала 7. Когда прочитывается маркер для ссылок, на ленте автоматически записывается маркер и адрес блока. После того, как адрес записан, машина записывает единицы в каналы информации вплоть до записи маркера. Каждый адрес, после того как он записан, прочитывается обратно, причем происходит прерывание блока маркера и происходит проверка программой. Единицы также считываются обратно и проверяются автоматически. Если обнаруживается ошибка, триггер ошибки адреса (разряд 24) устанавливается в "1" и после записи ведущих маркеров для следующего блока направление движения ленты меняется и маркеры для ссылок и маркеры блоков для ошибочного блока стираются. Разметка по адресам продолжается после ошибочного блока.

При переадресации ленты адреса ошибочных блоков вводятся в машину и производится поиск адреса блока на единицу большего чем ошибочный блок с наибольшим номером. Лента движется обратно и маркеры для ссылок и маркеры блока для ошибочных блоков стираются. После того, как последняя из этих операций проделана, лента останавливается и остающиеся блоки на ленте переадресуются. Выполняется процедура, аналогичная выполненной в программе разметки по адресам, но остающиеся маркеры не используются для указания присутствия исправных блоков.

Разряд указания ведущего адреса (разряд 34) необходим в программе переадресации для проверки на случай пропуска маркера для ссылок, в случае чего следующие маркеры начали бы рассматриваться как ведущие. Этот разряд устанавливается в "1" программой всегда, когда записывается ведущий адрес, и автоматически сбрасывается в "0" через отрезок времени от 10 до 50 мсек. Если он все еще находится в "1", когда достигается следующий маркер, лента и программа сбиваются с шага.

После завершения программы разметки или переадресации печатается команда оператору восстановить магнитофон "К" для нормальной работы. После того как это сделано, программа, в конце концов, записывает в первом блоке ленты число блоков информации, имеющихся на ленте, прежде чем вернуть управление системе управления.

5.2.9. СЧИТЫВАНИЕ МАГНИТНЫХ ЛЕНТ

ТИПА ОРИОН НА МАШИНЕ АТЛАС

Блок магнитной ленты, записанный на машине Атлас, содержит 512 48-разрядных слов. Блок, записанный на машине Орион, может содержать до 4096 слов. Поэтому на машине Атлас предусмотрено чтение лент, записанных на машине Орион.

Если блок менее чем в 512 слов, читается в направле-

нии вперед, обмен заканчивается обычным образом, когда прочтен следующий маркер блока. Это также дает сигнал для сравнения контрольной суммы, прочитанной с ленты, с суммой вычисленной по информации, прочитанной с ленты; соответствующий разряд контрольной суммы канала (разряд 24, строки 8-10 ЗУ "V" магнитной ленты) устанавливается в "0" или "1", смотря по тому, сходится ли результат или нет. Разряд "нет передачи 512 слов" (разряд 20, строки 8-15) также устанавливается в "1", но при этом не указывается, сколько слов прочтено.

Для блока более 512 слов фиксированная программа ЗУ должна устанавливать соответственно два или более регистра адреса страницы. Для обмена "вперед" счет происходит, как обычно, в разрядах 11-3 адреса обмена с лентой; после передачи 512 слов происходит "перенос" в разряды 14-12. Первые 512 слов прочитываются на первую страницу (т.е. страницу, у которой разряды 14-12 регистров установлены в 000), вторые 512 слов прочитываются на вторую страницу (т.е. разряды 14-12 регистра адреса страницы установлены в 001), и т.д., пока обмен не закончится. Если обмен не включает точного (нулевого) кратного 512 слов, устанавливается разряд "нет передачи 512 слов".

При команде "читать назад" производится аналогичная процедура с той разницей, что при чтении слов с ленты они передаются в регистр 511, 510, 509 и т.д. соответствующей страницы. Для передачи более 512 слов первая страница должна в разрядах 14-12 регистра адреса страницы иметь 111, вторая 110 и т.д. Программа фиксированного ЗУ в этом случае должна соответствующим образом изменить номера этих блоков.

Обмен считается законченным, и посылается соответствующий сигнал, когда прочтен ведущий маркер блока.

6.1. СТАНДАРТНАЯ КОМПИЛИРУЮЩАЯ ПРОГРАММА

Введение

Компьютерную подпрограмму машины "Атлас" можно рассматривать как автокод для написания автокодов. Более точно эта система дает возможность определить при помощи форматов утверждения и определений фразы форму утверждений (и их основных частей), которую следует использовать во входном языке. "Значение" этих утверждений или метод их перевода в конечный код описывается при помощи определений утверждений.

Данная система является самораспространяющейся; она дает возможность определить "значения" новых утверждений при помощи ранее определенных утверждений, а также при помощи основных утверждений ("значение" которых встроено в машину). Это также допускает рекурсивное определение как формы, так и значения утверждений. Это особенно полезно, например, при работе с алгебраическими выражениями, включающими любое количество скобок.

Маловероятно, что каждый оператор захочет писать свой собственный автокод, поэтому для более общих автокодов будут применимы стандартные определения. Определения автокодов Меркурия и Пегаса уже подготовлены. Был подготовлен вариант "Ввод и пуск" (Load and Go) Фортрана 709/7090; идет работа над полным вариантом Фортрана.

Проводятся предварительные изучения по языкам Алгол и Небула. (Небула - коммерческий язык, разрабатываемый фирмой Ферранти для вычислительной машины Орион). Оператор может использовать затем любой из этих языков автокодов или развить его за счет дополнительных определений для своей частной области. Если, однако, оператор выбирает некоторый другой язык ввода, тогда следует определить в этом языке форму и "значение" каждого утверждения. Хотя для определения "значения" других утверждений, имеются основные утверждения, их нельзя записать на языке ввода. Механизм для определения автокода описывается ниже со ссылкой на автокод Меркурия.

Определения фразы

Определения фразы используются для построения классов логически подобных фраз. Каждому классу присваивается имя, идентификатор класса, который можно затем использовать в дальнейших определениях фразы и форматах утвер-

ждения для указания, что в данном месте может стоять любая фраза рассматриваемого класса. Идентификаторы класса представлены последовательностью знаков, заключенных в квадратные скобки (напр., индекс можно приписать классу индексных букв). Поэтому следующие определения могут являться началом формального определения автокода Меркурия:

опред. фразы $[v] = a, b, c, d, e, f, g, h, u, v, w, x, y, z, \tau$.

опред. фразы $[v'] = a', b', c', d', e', f', g', h', u', v', w', x', y', z'$.

опред. фразы [индекс] = $i, j, k, l, m, n, o, p, q, r, s, t$.

опред. фразы [переменная] $[v], [v'], [v] [N], [v] [индекс], [v] [индекс] + [N]$.

Так же как класс $|N|$ обозначает целое число, он является встроенным классом и не требует дальнейшего определения, как например, класс $[K]$, обозначающий константу с плавающей запятой.

Две особенности, часто встречающиеся во фразе или утверждении, суть "повторные появления" и "необязательные появления" некоторой информации. Чтобы описать эти два случая, можно использовать указатели * и ?

Например:

$[A][B^*][C]$ означает $[A][B][C]$ или $[A][B][B][C]$ или $[A][B][B][B][C]$ и т.д.,

$[A][B][C]$ означает $[A][B][C]$ или $[A][C]$

а $[A][B^*?][C]$ означает $[A][C]$ или $[A][B][C]$ или $[A][B][B][C]$ и т.д.

В этих выражениях только $[A][B]$ и $[C]$ требуют формальных определений.

Общее арифметическое выражение в автокоде Меркурия может быть теперь определено следующим образом:

опред. фразы: $[Q] = [переменная], [K], [индекс]$

"-" : [терм] = $[Q +] [Q?]$

"-" : $[iQ] = / [Q]$

"-" : $[\pm] = +, -$

"-" : $[+ терм] = [+] [терм]$

"-" : $[общее выражение] = [\pm ?] [терм] [\pm терм * ?]$

Форматы утверждения

Форматы утверждений подобны определениям фразы за исключением того, что они описывают только одну фразу, а именно форму утверждения. Иногда удобно ввести утверждения, которые не входят в язык ввода и должны использоваться только при определении значения других утверждений. Этот тип утверждения выделяется написанием вспомогательного слова перед ним. Таким образом, форма класса арифметических команд в автокоде Меркурия определяется следующим образом:

формат утверждения: $[переменная] = [общее выражение]$ при определении "значения" этого класса команд (см. ниже) требуется несколько вспомогательных утверждений, включая следующие:

формат утв.(вспомогательный): накапл.сумм = [общее
выражение]
" - " " - " : [переменная] = накапл.
сумм
" - " " - " : накапл.сумм = [+ ?]
[терм]
" - " " - " : накапл.сумм = накапл.сумм
[+] [терм].

Определения утверждений

Каждому формату утверждения соответствует определенное утверждение. Первый описывает форму утверждения, а второе - действие, которое надо выполнить, когда эта форма встречается в исходной программе. В большинстве случаев это должно привести к комплектованию эквивалентного набора команд в конечной программе, но в случае декларативных утверждений, таких как "[V]" [N]" автокода Меркурия, эквивалентное действие состоит в том, что некоторая информация записывается в списки, используемые другими определениями утверждений. Утверждения могут, конечно, быть частично императивными, а частично декларативными, и обе операции в действительности выполняются при помощи команд составления списков (тип основного утверждения) с той единственной разницей, что в первом случае рассматриваемый "список есть конечная программа объекта.

Очень часто "значение" утверждения, в том смысле, как это было объяснено выше, можно выразить при помощи последовательности других менее сложных утверждений, где предвыражения главного утверждения есть параметры подутверждений. Поэтому необходимо иметь некоторые способы для выражения утверждения (и вообще выражения или фразы) через подвыражения, согласующиеся с их известной структурой, и если необходимо построить новые выражения, из этих подвыражений. Желательно также иметь возможность сравнивать различные выражения, чтобы выбрать правильные пути действия. По этим причинам введен второй тип основного утверждения, а именно операции обработки выражений.

Для осуществления "ветвлений" используется система плавающих адресов и любое подутверждение или основное утверждение, к которому должно передаваться управление, можно отметить;

Например,
 $IO \rightarrow 3$ если не [переменная] \equiv [V] [I] есть основное утверждение типа обработки выражений с отметкой IO.

Поэтому определение утверждения может состоять из трех типов "команд":

- | | |
|--|---|
| 1) подутверждений (определяются оператором); | } (основные утверждения, встроенные в систему); |
| 2) команд составления списков | |
| 3) операций обработки выражений | |

Прежде чем иллюстрировать примерами их использование, следует несколько пояснить команды составления списков.

Команды составления списков - с этими командами связана центральная группа 24-разрядных регистров, обозначенных $\alpha_1, \alpha_2, \alpha_3, \dots$, которые не образуют поля (т.е. которые нельзя рассматривать как α_2).

Кроме того имеется набор местных 24-разрядных регистров $\beta_1, \beta_2, \beta_3, \dots$, связанных с каждым определением утверждения. Команды составления списков касаются выборки, обработки и сравнения информации в данных регистрах и в регистрах, содержащих их адреса.

Так, например, запись $\alpha_{10} = \beta_2 + (\alpha_1 + 3)$

означает, что содержимое регистра IO равно содержимому регистра β_2 плюс содержимое регистра, адрес которого равен $\alpha_1 + 3$; а запись $\rightarrow I$ если $(\beta_1) > \alpha_3 + 2$ типична для различных проверок и означает "переход" к команде, отмеченной "I", если число в регистре, адрес которого находится в регистре β_1 , больше чем $\alpha_3 + 2$.

Вообще, если скажем α_1 есть адрес первой позиции в стандартном списке (т.е. в котором последовательные позиции помещены в регистры с последовательными адресами), то $(\alpha_1 + n)$ есть $(n + 1)$ позиция в этом списке. Кроме стандартного списка, однако, часто используется список цепочек. В этом списке каждое 24-разрядное слово, содержащее некоторую позицию, сопровождается (в следующем регистре) дополнительным словом, содержащим адрес следующей позиции (т.е. пары слов).

Вообще говоря адрес первого слова такого списка записывается в связующем слове последней позиции, таким образом, делая список замкнутым (циклическим). Преимущество списка цепочек состоит в том, что с ним легко оперировать, например, добавление и уничтожение позиций просто означает добавление или уничтожение связи. Однако, если в таком списке α_1 , скажем, является адресом первой позиции, то $(n + 1)$ -я позиция не будет теперь иметь адрес $(\alpha_1 + n)$. Вместо этого адрес второй позиции фактически определяется как, скажем, $\beta_1 = (\alpha_1 + 1)$, а третьей, как $\beta_2 = (\beta_1 + 1)$, и т.д. Для краткости обращения к различным позициям цепочки используется символ θ ; $(n + 1)$ -я позиция обозначается таким образом через $(\alpha_1 \theta_1)$ и эту позицию можно передать в β_5 при помощи команды $\beta_5 = (\alpha_1 \theta_n)$. Таким образом $\alpha_1 \theta_1$ эквивалентно $(\alpha_1 + 1)$, если α_1 есть адрес в цепочке. Использование команд составления списка и операций обработки выражений и подутверждений можно проиллюстрировать следующими примерами определения утверждений:

опред. утв. : [переменная] = [общее выражение]
 накапл.сумм. = [общее выражение]
 [переменная] = накапл.сумм.
 конец

Первая позиция в каждом определении утверждения - есть заголовок утверждения. Он служит как для указания на связь определения, в котором он появляется, с конкретной формой утверждения, так и для установления места

нахождения подвыражений (в приведенном случае [переменная] и [общее выражение]) в утверждении, которое можно записать как параметр последующей "команды". В приведенном примере последующие "команды" являются просто подутверждениями, но вообще параметры, появляющиеся в заголовке утверждений, нельзя отождествлять непосредственно с подутверждениями, и они должны быть расписаны в более элементарных выражениях посредством команд класса обработки параметров.

опред. утвержд.: накапл. сумм. = [общее выражение]
 пусть [общее выражение] = [+ ?][терм] [± терм* ?]
 накапл. сумм. = [+ ?] [терм]
 1 → 1 если [+ терм* ?] не ≡ [+ терм*]
 3 → 2 если [± терм*] не ≡ [± терм] [± терм*]
 пусть [+ терм] = [+] [терм]
 накапл. сумм. = накапл. сумм [±] [терм]
 → 3
 2] пусть [± терм] ≡ [+] [терм]
 накапл. сумм. = накапл. сумм [±] [терм]
 1] конец

В этом примере только один параметр входит в заголовок утверждения, поэтому очевидно, что если это утверждение должно определяться при помощи более простых утверждений, то это параметрическое выражение должно разрешаться в подвыражениях. Это выполняет первая "команда" определения, затем можно обращаться к выражениям в первой части равенств. Следующая "команда" есть подутверждение, которое здесь не определяется, но согласно которому первый член общего выражения (возможно с тем или другим знаком) будет введен в накапливающий сумматор. Затем должен быть определен характер члена [+ терм* ?], так как это может быть или [+ терм*] или "нуль". В этом последнем случае управление переходит к концу посредством команды условного разрешения параметров. Если управление переходит последовательно к следующей "команде", то [общее выражение] должно включать подвыражение [+ терм*]. Теперь для решения его в более элементарной форме необходимо знать, как определяются внутри машины классы " * ". В общем случае [идентификатор*] определяется рекурсивно как [идентификатор] [идентификатор*], [идентификатор]. Рекурсивная структура подвыражения [+ терм*] раскрывается циклом команд, которые обрабатывают каждое подвыражение [+ терм] в последовательности до самого последнего, который имеется в пункте, отмеченном " 2 ". Как и при обычном программировании, можно видеть, что одно и то же название, в данном случае [+ терм*], можно динамически приписать последовательности различных выражений.

Хотя вышеприведенные примеры не содержат основных команд составления списков, очевидно, что определения некоторых использованных подутверждений приведут, в конце концов, к последовательностям основных команд составления списков, которые будут компилировать конечную программу. Некоторые из этих определений утверждения более низкого уровня потребуют знания различных указаний (т.е. утверждений типа [V] [N]), встречающихся ранее.

Один из способов получения этой информации — это записать для определения утверждения, связанного с [V] → [N], [N], связанные с каждой из букв определителя фразы [V] в конкретном положении в стандартном списке из 15 регистров, отдельно от конечной программы. Очевидный способ связи букв определительной фразы [V] с положением в списке состоит в использовании того же порядка, что и в определении фразы [V], т.е. "а" в положении 1, "в" в положении 2 и т.д. Однако при этом требуется, чтобы определенным образом предусматривалось определение, какой именно случай из определения фразы представляет данное выражение. Для этой цели предназначена встроенная команда.

[α] = категория ["идентификатор любой фразы"]
 Таким образом в приведенном ниже примере всякий раз, когда рассматриваемое [V] есть, скажем "α", величина β₁ становится равным 4. Ниже предполагается, что α₃ было остановлено для адреса списка указаний и что оно не изменится другими определениями утверждения, но последние могут, конечно, обращаться к нему.

опред. утверждения: [V] → [N]
 β₁ = категория [V]
 β₁ = β₁ - 1
 (α₃ + β₁) = [N]
 конец

Заключение

Очевидно, приведенное выше краткое описание не является достаточным, поэтому для ознакомления с более полным описанием читатель отсылается к работам, приведенным ниже. Однако это краткое описание иллюстрирует относительную легкость, с которой могут быть составлены компилирующие программы для конкретных программирующих языков.

Литература

1. An Assembly Programme for a Phrase Structure Language, Computer Journal, October 1960.
2. An Assembly Programme for a Phrase Structure Language /concluded/, Computer Journal, January 1961.
3. A Description of Mercury Autocode in terms of a Phrase Structure Language, Annual Review in Automatic Programming, 1961.

6.2. ПРОМЕЖУТОЧНАЯ КОМПЛЕКТУЮЩАЯ ПОДПРОГРАММА

Общее описание

Эта подпрограмма в основном предназначена для использования на то время, пока разрабатывается главная компилирующая программа. Она будет записана в ферритовой памяти, хотя (по крайней мере в начале) может храниться и на магнитном барабане. В главной компилирующей программе будет учтена эта подпрограмма и, таким образом, программы, написанные при помощи нее, также могут позже использоваться. Принятые соглашения таковы, что все использованные знаки употребляются в выводном устройстве Пегаса и Меркурия на пятидорожечной ленте, а также на стандартном семидорожечном флексорайторе.

Пробивается команда F, Va, Vm, S, где F состоит из одного двоичного и трех восьмеричных разрядов. Если нужно, то нули в левой части F можно опустить. Va, Vm находятся в ячейках от 0 до +127 и состоят из трех десятичных разрядов.

S является адрес.

Каждая часть команды оканчивается запятой, а полная команда оканчивается возвратом каретки и подачей строки.

Адреса

Адрес может быть десятичным, восьмеричным, плавающим или общим, т.е. комбинацией трех других.

Десятичный адрес может быть записан: (a) как знак (необязательно, если он положителен), за которым идет ряд десятичных разрядов, напр., 31913 или + 976 или -7928. В этом случае предполагается, что двоичная запятая числа в вычислительной машине отстоит на три двоичных разряда (или один восьмеричный) от конца числа с младшими разрядами (b) как и в пункте (a), но в конце стоит запятая и одна восьмеричная цифра, например: ,

976,7 или - 325,1.

В этом случае восьмеричная цифра входит в счет трех младших двоичных разрядов числа.

(c) как и в пункте (a), но с символом " n ". Этот символ можно пробить в любом месте адреса, но рекомендуется пробивать его или непосредственно перед первым, или непосредственно после последнего десятичного разряда, например, n 736, - n 219, -514n, + n99. В этом случае число в машине хранится с двоичной запятой после младшего разряда. Влияние " n ", таким образом, состоит в сдвиге числа в строку младших разря-

дов на один восьмеричный знак (допускается только один такой сдвиг).

Восьмеричный адрес пробивается (d) как звездочка (*), за которой следуют до восьми восьмеричных разрядов. Первая пробитая цифра сдвигается в старший восьмеричный разряд числа, состоящего из 24 двоичных разрядов, вторая цифра - следующий восьмеричный разряд и т.п.

Нули, стоящие перед старшим разрядом числа, можно при желании опустить.

Плавающий адрес пробивается (e) как ряд десятичных цифр, заключенных в скобки, напр. (37). Он может быть определен либо при помощи управления (указание строки), либо при помощи отметки команды (или константы). Отметка записывается на той же строке, что и команда и оканчивается правой скобкой, напр. (21) F, Va, Vm, S.

Когда плавающий адрес определяется уравнением, отметка находится в левой части уравнения, а общий адрес - в правой.

Возможно также установить отметку, равную другой, напр.

$$\begin{aligned} (37) &= \text{адрес} \\ (37) &= (45). \end{aligned}$$

Отметка 0 есть адрес выполняемой команды; невозможно установить отметку 0, отмечая команду. Ее можно, однако, установить указанием строки. Перед отметкой не может стоять минус.

Общий адрес пробивается или (f) как восьмеричный адрес (d), которому предшествует десятичный адрес / (a), (b) или (c) /, например,

$$\begin{aligned} + 80,6 * 216 \\ - n 27 * 53. \end{aligned}$$

Конечный результат - есть сумма двух частей: восьмеричное представление вышеприведенных двух примеров есть 21601206 и 527777 45, соответственно; или (g) как ряд отметок (e), которые могут иметь адреса в виде (a), (b), (c), (d) или (f), в начале, конце или между любыми двумя членами цепочки. Окончательный адрес - есть сумма составляющих частей, например,

$$- 2.4(4)6 * 412(3) - 3n.$$

Если отметки (3) и (4) были предварительно определены уравнениями

$$\begin{aligned} (3) &= *017 \\ (4) &= *000211. \end{aligned}$$

Тогда этот адрес (в восьмеричной системе) есть 43121131. Частный пример этого типа - есть относительный адрес, например, I(0) - есть адрес команды, непосредственно идущей за данным адресом.

2(0) - есть адрес следующей команды и т.д.

- I(0) - есть адрес команды, непосредственно предшествующей данной.

Адрес хранится в регистре из 24 двоичных разрядов; два "адреса" могут быть на одной печатной строке перфоленты, разделенные наклонным штрихом. Таким образом, последовательность "адресов" считывается в виде последовательности полуслов в 24 двоичных разряда каждое, если они пробиваются попарно, как было описано выше.

Указания

Указание предусматривает очистку отметок от I до n и это печатается в виде:

" Очистить n отметок ",
где n может быть целым числом или общим адресом.
Указание для записи программы пробивается как ЗАПИСЬ,
за которым идет общий адрес.

Ссылки

- 1) немедленно, если отметка уже установлена;
- 2) при указании ВВОД;
- 3) при выполнении указания ОЧИСТИТЬ n ОТМЕТОК
- 4) когда собраны все 412 ссылок.

В этом последнем случае, если некоторые отметки не были установлены, соответствующие ссылки продолжают храниться в памяти.

Числа с плавающей запятой

Число с плавающей запятой из 48 двоичных разрядов можно пробить в форме числа с фиксированной или плавающей запятой. Если оно пробивается в виде числа с плавающей запятой, то мантисса числа пробивается перед порядком его и отделяется запятой. Для целых чисел десятичная запятая не пробивается, а знак плюс может быть опущен как для положительных мантисс, так и для положительных порядков числа. Число оканчивается возвратом каретки и подачей строки.

Например,
$$\begin{array}{r} 123, \quad -2 \\ - 147. \quad 631 \\ + \quad 77 \\ \hline 1973, 5, 1 *) \end{array}$$

Знаки, не принимаемые во внимание

Не принимаются во внимание знаки стирание, пробел и подача строки на пятидорожечной ленте.

Знак сдвиг цифры не принимается во внимание, если он пробивается, когда выводное ленточное оборудование находится в положении сдвига цифры.

Знак сдвиг буквы не принимается во внимание, если он пробивается, когда выводное ленточное оборудование находится в положении сдвига буквы.

*) Точка эквивалентна десятичной запятой. Прим. ред.

Индикация ошибки

Указание печатается, если обнаружена ошибка в пробивке, например, две запятые после какой-нибудь части команды; где возможно, остаток программы считывается и обрабатывается. Программа отбрасывается, если при достижении указания ВВОД не установлены какие-нибудь отметки, на которые уже встречались ссылки. В некоторых кодах A, где компонента операции, хранящаяся в памяти не определена командой, а также в неразмещенных В кодах типа S, разряды S должны содержать адрес фиксированной памяти (т.е. адрес, старший восьмеричный разряд которого равен четырем) иначе указывается ошибка (8). Выявленные ошибки делятся на 8 групп, при обнаружении ошибки печатается номер соответствующей группы.

Типы ошибок:

- 1) ложный знак;
- 2) неправильный формат;
- 3) отметка, установленная дважды;
- 4) не установленная отметка;
- 5) слишком большое количество ссылок на дальнейшую информацию;
- 6) ошибка в написании указания;
- 7) неопознанный код;
- 8) неправильный адрес.

Раздел 7 - КОДЫ БУМАЖНОЙ ЛЕНТЫ, ПЕРФОКАРТ
И ПЕЧАТАЮЩЕГО УСТРОЙСТВА

7.1. КОДЫ БУМАЖНОЙ ЛЕНТЫ

7.1.1. КОД ПЯТИДОРОЖЕЧНОЙ БУМАЖНОЙ ЛЕНТЫ

Это тот же самый код, который употребляется в выводном устройстве "Крид" (creed) для вычислительных машин Пегас и Меркурий. Перевод кода бумажной ленты в код вычислительной машины выполняется программой, хранящейся в фиксированной памяти.

Знак на ленте	Печатаемый знак	
	Сдвиг в регистр букв	Сдвиг в регистр знаков
---	сдвиг в регистр цифр	
---	A	1
---	B	2
---	C	*
---	D	4
---	E	(
---	F)
---	G	7
---	H	8
---	I	≠
---	J	=
---	K	-
---	L	∨
---	M	подача строки
---	N	пробел
---	O	0
---	P	∨
---	Q	∨
---	R	∨
---	S	∨
---	T	∨
---	U	5
---	V	6
---	W	/
---	X	x
---	Y	9
---	Z	+
---		сдвиг в регистр букв

0 0 0 : - -
0 0 0 : - 0
0 0 0 : 0 -
0 0 0 : 0 0

и ?

п
возврат каретки
* (стирание)

7.1.2. КОД СЕМИДОРОЖЕЧНОЙ БУМАЖНОЙ ЛЕНТЫ

Для печати и считывания знаков с семидорожечной бумажной ленты будет использоваться флексорайтер. Был выбран такой код, что пробитые знаки всегда имеют нечетное число отверстий; в этом случае число принимается печатающим устройством. Дорожка P4 используется для обозначения четности *. Программа, хранящаяся в фиксированной памяти при вводе превращает ленточный код в машинный, при этом осуществляется табличный контроль того, что считываемый знак имеет нечетное число пробивок. На печатающих рычагах (а также на соответствующих ключах) имеется два набора различных знаков в группе I, как показано ниже.

Знак		Знак на ленте
Верхний регистр	Нижний регистр	
Группа 0		
0	пробел	-- 0 - . - - -
1		
2	новая строка	-- - - . - 0 -
3	поворот бумаги	-- 0 - . - 0 0
4	таблица	-- - - . 0 - -
5	сдвиг влево (возврат)	-- 0 - . 0 - 0
6	нижний регистр	-- 0 - . 0 0 -
7	верхний регистр	-- - - . 0 0 0
8		
9		
10		
11		
12	останов	-- 0 0 . 0 - -
13	перфорирование	-- - 0 . 0 - 0
14	считывание	-- - 0 . 0 0 -
15	:	-- 0 0 . 0 0 0
Группа I		
0	' (апостроф)	- 0 - - . - - -
1	1	- 0 0 - . - - 0
2	1	- 0 0 - . - 0 -
3	∨	- 0 - - . - 0 0

*) Третья дорожка слева в приводимой ниже таблице.
Прим.ред.

4	4	>	-	0	0	-	.	0	-	-
5	5	=	-	0	-	-	.	0	-	0
6	6	(подчеркивание)	-	0	-	-	.	0	0	-
77	7	{	-	0	0	-	.	0	0	0
8	8	}	-	0	0	0	.	-	-	-
9	9)	-	0	-	0	.	-	-	0
10	α (IO)	π (L)	-	0	-	0	.	-	0	-
11	β (II)	-1 (%)	-	0	0	0	.	-	0	0
12	1/2	?	-	0	-	0	.	0	-	-
13	+	&	-	0	0	0	.	0	-	0
14	-	(звездочка)	-	0	0	0	.	0	0	-
15	.	,	-	0	-	0	.	0	0	0

Группа 2

0	A	a	0	-	0	-	.	-	-	0
1	B	b	0	0	-	-	.	-	0	-
2	C	c	0	-	-	-	.	-	0	0
3	D	d	0	-	0	-	.	0	-	-
4	E	e	0	-	-	-	.	0	-	0
5	F	f	0	-	-	-	.	0	0	-
6	G	g	0	-	0	-	.	0	0	0
7	H	h	0	-	0	0	.	-	-	-
8	I	i	0	-	-	0	.	-	-	0
9	J	j	0	-	-	0	.	-	0	-
10	K	k	0	-	0	0	.	-	0	0
11	L	l	0	-	-	0	.	0	-	-
12	M	m	0	-	0	0	.	0	-	0
13	N	n	0	-	0	0	.	0	0	-
14	O	o	0	-	0	0	.	0	0	0
15			0	-	0	0	.	0	0	0

Группа 3

0	P	p	0	0	0	-	.	-	-	-
1	Q	q	0	0	-	-	.	-	-	0
2	R	r	0	0	-	-	.	-	-	0
3	S	s	0	0	0	-	.	-	0	0
4	T	t	0	0	-	-	.	0	-	-
5	U	u	0	0	0	-	.	0	-	0
6	V	v	0	0	0	-	.	0	0	-
7	W	w	0	0	-	-	.	0	0	0
8	X	x	0	0	-	0	.	-	-	-
9	Y	y	0	0	0	0	.	-	-	0
10	Z	z	0	0	0	0	.	-	0	-
11							.			
12							.			
13							.			
14							.			
15							.			

* (стирание)

Раздел 9 - ТЕСТОВЫЕ ПРОГРАММЫ

9.1. ТЕСТЫ ДЛЯ ЦЕНТРАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ
МАШИНЫ

Тесты будут делиться на ряд отдельных тестовых программ, известных под названием тест-программы (Т.Р.С.)
Список некоторых возможных отдельных тест-программ:
Хранение и выборка строки В из В-регистров В, s, В, n арифметических операций.

Тесты В

Передача в накапливающий сумматор, сложение, вычитание и соответствующие команды перехода с фиксированной запятой.
Умножение в накапливающем сумматоре (с фиксированной запятой).
Передача в накапливающий сумматор, сложение, вычитание и соответствующие переходы с плавающей запятой.
Умножение в накапливающем сумматоре (с плавающей запятой).
Деление в накапливающем сумматоре (с плавающей запятой).
Проверка запоминающих устройств.
Проверка барабана.
Проверка функций магнитной ленты.
Проверка работы магнитной ленты.
Тесты магнитных лент, т.е. записать и считывать с ленты.
Проверка различных входных устройств.
Проверка различных выходных устройств.
Тест-программы для главной машины будут храниться по возможности в постоянной памяти. Ведущая программа управляет операциями и связями тест-программы. Тест-программы главной машины могут выполняться по отдельности или последовательно. По выполнении одной тест-программы управление передается вновь ведущей программе и делается короткая проверка из команд, используемых при контроле. Данная тест-программа вводится 10 раз, затем выполняются следующие, или повторяется та же самая, в зависимости от записи в ведущей программе.
Компоненты операций для проверочных команд и ожидаемый результат хранятся в последовательных ячейках. Необходимы специальные В-регистры, для того чтобы при помощи модификатора выбрать действительные адреса чисел, участвующих в операции (компонентов операций).

В некоторых случаях В-регистры устанавливаются в 0.

Где возможно, используются следующие команды при проверке результатов:

$ba' = s,$
 $ba' = ba \neq s,$
 $ba' = n,$ если $bm \neq 0$

В каждой тест-программе имеется последовательность команд следующего типа:

s проверка ЗУ
s + 8 переход к подпрограмме "анализ ошибок"
s + 16 {
s + 24 { 2 псевдокоманды

Эти 2 псевдокоманды содержат следующую информацию:

Номер тест-программы.

Номер операции.

Адрес ожидаемого результата.

Адрес в тест-программе, к которому возвращается управление.

Один двоичный разряд для указания печатать или нет специальную информацию.

Возможно 1 двоичный разряд для определения, печатать или нет информацию из регистров В.

Специальная информация находится в форме слова, например СДВИГ, и хранится непосредственно после результата (т.е. первая компонента операции Специальная информация).

Номер операции может храниться двумя способами, один — как он хранится в машине и второй — для целей печатания.

Подпрограмма "анализ ошибки" должна включать возможно меньше различных команд; операции также должны быть возможно проще. Эти требования могут удовлетворяться за счет использования лишних команд.

Когда появляется ошибка, начинает работать подпрограмма "анализ ошибки". Для многих команд, включающих две компоненты операций, имеет место следующая операция. Команды, используемые при проверке на ошибку, проверяются, скажем, 10 раз и только если они оказываются правильными выполняется следующая операция, в связи с предполагаемой ошибкой. Эта операция состоит в том, что сначала печатается функциональный номер ошибки, возможно номер тест-программы и специальная информация, если это указано, затем находится вероятность появления ошибки (в процентах) при 50-кратной проверке команды. Этот процент печатается, и если он меньше 50, то дальнейший анализ не проводится, и уравнение передается вновь тест-программе. В противном случае две компоненты операций и ожидаемый результат печатается в двоичном коде. Печатается, столько, сколько раз какой-либо разряд содержит неверную цифру, начиная со старших разрядов. Перед печатью данных по каждому разряду печатается номер этого разряда. Команда выполняется тогда циклически; выйти из цикла возможно лишь путем изменения управления вручную.

Подпрограмма анализа ошибки должна быть рассчитана на три основных типа команд:

Команды для накапливающего сумматора (с фиксированной запятой).

Команды для накапливающего сумматора (с плавающей запятой).

Команды для регистров В.

Некоторые особенности деления, умножения и сложения включают четвертый тип.

Для команд, не подходящих для подпрограммы анализа ошибки, описанной выше, будет печататься только функциональный номер ошибки перед тем, как управление передается вновь в тест-программе. (Возможно будет также печататься специальная информация из регистров В).

Единственной особенностью команд для накапливающего сумматора, включающих две компоненты операции, которые не подходят для анализа ошибки, есть деление с удвоенным количеством разрядов, так как для этого требуется первоначально заполнить весь накапливающий сумматор. Любая команда, приводящая к тому, что в накапливающем сумматоре появляются как старшая, так и младшая половины результата, проверяется по частям.

Из трех основных типов операций с В-регистрами для проверки арифметических операций используется операция типа " $ba' = ba$ с s ". Команды этого типа подходят для анализа ошибки. Проверяется только дешифрование других типов и используется упрощенная подпрограмма анализа ошибки. Нормальный переход к тест-программе осуществляется в начале предварительной программы. Команды предварительных испытаний, используемые в ведущей программе и программе анализа ошибки, начинаются с определенных переходов и некоторых арифметических операций с В-регистрами. Если какая-либо команда появляется неправильно, начинает работать простая программа печати, а предварительная программа начинается опять сначала, за исключением случая, когда $B \geq 0$ находится в B_0 и когда неизвестно, какой переход работает. Предварительная программа выполняется правильно 10 раз, затем печатаются числа от 0 до 9. Затем включается ведущая программа и циклически выполняются тест-программы от 1 до n . Если требуется выполнить одну отдельную тест-программу, управление устанавливается на соответствующей позиции в списке точек входа, затем вводится тест-программа и непрерывно выполняется. Порядок тестовых команд в предварительной программе следующий:

Тест $B \neq 0$ для B_0

Тест $B \geq 0$ для управления

Тест $B \geq 0$ для B_0

Тест $B \neq 0$ для управления

$b' = n$ (изменение В-регистров)

$b' = n$ или $b' = b \neq s$ для того же числа

$b' = b \neq s$ для разных чисел

$b' = s$

$s' = b$

$b' = b + n$ (операции сложения, используемые в программе анализа ошибки для вычисления и сложения с модификаторами).

Простая модификация.

В программе анализа ошибки и в ведущей программе счет ведется от $-n$ до нуля; а модификаторы первоначально положительны и возрастают. Эти методы применяются для того, чтобы использовать сумматор В возможно реже.

Раздел 11 - ОПИСАНИЕ ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ (MUSE)

11.1. СТРУКТУРА МАШИНЫ

Вычислительная машина "Атлас" включает :

- постоянное ЗУ емкостью 8192 слова
 - рабочее ЗУ емкостью 1024 слова
 - ЗУ на сердечниках емкостью 16384 слова
 - 4 барабана (M.D.S.)
 - 8 приспособлений для магнитных лент (Amrex FR 300)
 - 1 считывающее устройство на перфокартах со скоростью считывания 600 карт в минуту (I.C.T.)
 - 1 перфоратор (I.C.T. 582)
 - 1 печатающее устройство на 400 строк в минуту (I.C.T.)
 - 4 печатающих устройства "Flexowriter"
 - 4 телетайпа
 - 4 устройства считывания с лент (TR5)
 - 1 графикопостроитель
 - 10 автономных устройства типа "Flexowriter".
-

Раздел I3 - ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО "v"

I3.1. АДРЕСАЦИЯ ЗУ "v"

Положение различных частей памяти "v" указывается в адресе разрядами I4, I3 и I2 следующим образом:

I4	I3	I2	разряды положения
0	0	0	Центральная вычислительная машина
0	0	I	память на магнитных сердечниках
0	I	0	барабан
0	I	I	магнитная лента
I	0	0	внешнее оборудование (исключая ленты и барабаны)

Обозначения, используемые в следующих разделах для указания считывания или записи в разряды памяти V, содержат:

- R - там, где возможно только считывание; запись не производится и разряды, помеченные другими символами, всегда читаются как нули
- WI - там, где возможна только запись единиц; чтение дает нули, а запись нулей не выполняется.
- WIO - там, где возможна запись и единиц, и нулей; чтение всегда дает нули
- RWI - там, где возможно чтение, а также запись единиц; запись нулей не выполняется
- RWIO - там, где возможно чтение и запись нулей и единиц

I3.2. ЗУ v ЦЕНТРАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ

Первые три строчки связаны с прерыванием и используются для определения источника прерывания

Регистры	Разряды	
0	3I RWI	Контрольный разряд I-для оперативной памяти
	30 RWI	Контрольный разряд 2-для обращения к барабану. Этот разряд устанавливается при обнаружении ошибки при записи и при считывании
	29 RWI	Контрольный разряд 3. Обращение к магнитной ленте. Этот разряд устанавливается, если выявляется ошибка

	28	RWI	(по четности) в содержании страницы оперативной памяти, записываемой на магнитную ленту
	27	RWI	Контрольный разряд 4 - для вспомогательной памяти
I	3I	R	Контрольный разряд 5 - для постоянной памяти
	30	R	Барабаны. Логическая операция "или" разрядов в регистре, связанном с барабаном памяти v
	29	RWI	Неисправность коммутатора магнитных лент. Логическая операция "или" разрядов регистра 27 памяти v, относящейся к магнитным лентам
	28	RWI	Переполнение порядка в накапливающем сумматоре
	27	RWI	Постраничная блокировка. Разряд устанавливается при обращении программы к заблокированной странице оперативной памяти
	26	RWI	Контрольный разряд устанавливается при обращении программы к "запрещенной" (резервированной для других целей) части памяти
	25	RWI	Контрольный разряд устанавливается при выполнении команды, взятой из "запрещенной" части памяти
	24	RWI	Деление на ноль или ненормализованное число (устанавливается только операциями 374, 376, 377).
2	3I	R	Разряд устанавливается при обнаружении кодов непредусмотренных операций
	30	R	Контрольный разряд - логическая операция "ИЛИ" разрядов регистра "0"
	29	R	Внешние устройства P1 (устройства считывания с перфокарт)
	28	R	Маркеры блоков магнитных лент. Логическая операция "или" строки 26 памяти v на магнитной ленте
	27	R	Внешнее устройство P2 (печатающие устройства, графические выводные устройства, контрольные устройства TR7), ксерографические печатающие устройства
	26	R	Внешние устройства P3 (телетайпы, быстродействующие перфораторы и др.)
	25	RW-I	Внешние устройства P4 (счетчик команд, синхронизаторы, телепринтеры и флексорайтеры, перфораторы)
	24	R	Отсутствие адреса данной страницы в регистрах адресов страниц
			Логическая операция "или" разрядов регистра I

3	25	RWIO	I/ME - контрольный триггер устанавливается в "I", если имеется I.
	24	RWIO	II запрещение прерывания (устанавливается в "I" для запрещения прерывания)
4	24	RWIO	M/E - контрольный триггер (устанавливается в "I", если M)
6	29	RWIO	I2/I3 Сдвиг. "I", если сдвиг I2 устанавливается в ноль при каждом делении
	28	RWIO	QS знак частного в основных командах деления
	27	RWIO	AO (Фиксированная запятая), (переполнение сумматора)
	26	RWIO	} проверка В-регистра
	25	RWIO	
	24	RWIO	Bt ≠ 0
7	31-24	R	Вс перенос в регистрах В 8 тумблеров

13.3. ЗУ v НА МАГНИТНЫХ СЕРДЕЧНИКАХ

Регистры	Разряды	
0-31	WIO	47 46-36
	WIO	
32	R	47-32
33	R	47-32
34	R	47
	R	46-36
	R	35

Блокировка страниц } Регистры
Адрес блока } адресов
страниц

Разряды "использования" страниц с I5 по 0
Разряды "использования" страниц от 3I до I6

Не принимать во внимание блокировку, т.е. неэквивалентность, возникающую при обмене с барабаном, лентой или когда машина находится на управлении прерывания

Неэквивалентный адрес блока
Неэквивалентность операнда

Вышеуказанное распределение разрядов пригодно для "Атласа" с 32 страницами памяти на сердечниках: для большей памяти на сердечнике разряды "использования" поместятся на следующих регистрах после последнего регистра адресов.

Разряды на страницах 32, 33, 34 сбрасываются в ноль автоматически, когда с них считывается информация (программой).

Разряд "не принимать во внимание блокировку" устанавливается в I при каждом обращении к памяти на сердечниках, при котором получается прерывание по неэквивалентности во время обращения к барабану или ленте,

или когда машина находится на управлении прерывания. Он устанавливается в ноль для нормальных прерываний. Адрес "неэквивалентного" блока устанавливается, когда происходит прерывание из-за того, что требуемая команда или операнд недоступны непосредственно в памяти на сердечниках, т.е. блок необходимой информации может или отсутствовать в памяти на сердечниках, или может быть заблокирован.

Разряд неэквивалентности операнда устанавливается в I, когда прерывание неэквивалентности появляется из-за того, что требуемый операнд не находится в памяти.

13.4. ЗУ v НА БАРАБАНАХ

(Максимум 32 быстрых барабана, тип МД5)

Регистр	Разряд	
0-31	R	29-27
	R	26
32	R	29-27
36	WIO	29-27
40	WIO	29-27
44	WIO	28-26
48	WIO	27-26
52	WIO	29
	WIO	28-26
56	WI	29
	WI	28
	WI	27
	WI	26
60	RWI	28
	RWI	27
	RWI	26

0 регистры
"2-II/2" разряд
(для барабанов 0-31 соответственно)
Заполнение блоков.
Количество требуемых блоков
Начальная "0"
Группа дорожек на соответствующем барабане
№ барабана (внутри стойки)
Быстродействующий барабан (0/I)
№ стойки (шкафа)
Начало
Смена выборки
Чтение
Запись
Отсутствие стойки с барабанами
Ошибка в передаче на барабан
Окончание передачи на барабан

Логическая операция "ИИИ" разрядов регистра 60 устанавливает разряд 3I регистра I памяти v для Центральной вычислительной машины.

Регистр 56 содержит управляющие разряды.

13.5. ЗУ v НА МАГНИТНЫХ ЛЕНТАХ

Регистр	Разряд	
0-6	36-24	R
7	36-24	RWIO

Регистр текущего адреса данного блока для каналов 0-6
Регистр текущего адреса данного блока для канала 7

8-15	41-24		Регистры команд ленты	
	41	RWI	Большая скорость	
	40	WI	Нормальная скорость	
	39	RWI	Обратное направление	
	38	WI	Прямое направление	
	37	RWI	Пуск	
	36	WI	Останов	
	33	RWI	Чтение с регенерацией	
	32	WI	Нормальное чтение	
	31	RWI	Считывание в следующем блоке	
	30	WI	Конец считывания в следующем блоке	
	29	RWI	Запись	
	28	WI	Конец обмена	
	27	RW	Разрешение записи	
	26	R	Конец ленты	
	25	R	Передача неполной страницы	
	24	RWI	Ошибка в контрольной сумме. Запись "1" снова гасит разряд	
16-23	28-24	WIO	Номер магнитофона для каналов 0-7	
	24	39-24	RWI	Триггеры занятости для магнитофонов 15-0
	25	39-24	RWI	Триггеры занятости для магнитофонов 31-16
	26	31-24	RWI	Прерывание по адресу блока для каналов 7-0 Запись "1" устанавливает разряд в "0".
	27	31-24	RWI	Прерывание из-за неисправности магнитофонов для каналов 7-0. Запись "1" гасит разряд в "0".
	28	35-24		Регистр управления адресом ленты только для канала 7
	35	R	Изменение № магнитофона	
	34	RWI	Указание ведущего адреса	
	33	RWI	Разрешение счета	
	32	WI	Запрещение счета	
	31	WI	Запись маркера обращений	
	30	WI	Не записывать маркер обращений	
	29	WI	Записать "1"	
	28	WI	Не записывать "1"	
	27	WI	Присвоить адрес ленте	
	26	WI	Не присвоить адрес ленте	
	25	RWI	Ошибка в адресе	
	24	WI	Снова установить сигнал ошибки в адресе.	

Разряд 28 регистра 2 памяти V, относящейся к Центральной вычислительной машине, хранит результат логической операции "ИЛИ" разрядов регистра 26. Аналогичную роль играет 30-й разряд I регистра по отношению к 27 регистру.

Память V для ленты включает регистры в самих магнитофонах. Некоторые из них являются триггерами типа

внешних устройств, требующими длительных импульсов запуска. Необходимость генерации таких импульсов увеличивает время обращения к памяти V на лентах до 5 мксек.

13.6. ВНЕШНЕЕ ЗУ V

13.6.1. УПРАВЛЯЮЩИЕ РАЗРЯДЫ И РАЗРЯДЫ ИНФОРМАЦИИ

(а) Разряды информации

Для устройства ввода необходимо только считывание информации с триггеров и не предусмотрена запись в них с вычислительной машины (например, $s' = b$ не влияет на триггеры информации).

Для устройства вывода необходимо только записывать в триггерах информации и не предусмотрено считывание вычислительной машиной (например, $s' = b$ устанавливает разряды регистра V в ноль, когда есть регистр, содержащий разряды информации в устройстве вывода).

Разряды информации снова устанавливаются автоматически в ноль после вывода информации. Команды записи (например, $s' = b$) записывают только единицы; попытки записать нулей не оказывают действия и триггеры остаются в прежнем состоянии.

(в) Управляющие разряды: незанятость, пуск, сигнальный

Для разрядов "незанятости" и "пуска" обычно необходимо, чтобы вычислительная машина могла считать состояние нужного триггера, установить триггер и погасить его. Триггер устанавливается путем записи единицы в разряд памяти V; запись ноля не произведет действия. Триггер гасится путем записи единицы в другой разряд памяти V. Таким образом, если установлен триггер "пуска", запись единицы в разряд "останова" сбросит этот триггер. То же верно для разрядов "незанятости". Поскольку два соответствующих разряда находятся в одном регистре памяти V (см. ниже), вычислительная машина записывает оба разряда в одной команде, т.е.

0	0	не производит действия, триггеры остаются в прежнем состоянии
0	1	устанавливается триггер (например, "незанятость")
1	0	гасит триггер (например, "занятость")

Программы в постоянной памяти будут записываться так, чтобы комбинация "11" не возникала, так как при

этом триггер может установиться в неопределенном положении (либо 0, либо 1). Что касается "сигнальных" разрядов, триггер считывается в тот регистр памяти V, где сгруппированы вместе восемь таких разрядов. Эти разряды устанавливаются автоматически, внешними устройствами, и, следовательно, необязательно устанавливать их вычислительной машиной. Они гасятся путем записи единицы в разряд другого регистра памяти V.

Используются следующие обозначения:

- R - там, где возможно только считывание, а запись не действует
- W - там, где возможно только для записи единиц; чтение дает всегда нули, а запись нулей не производится
- * - там, где возможно считывание, а также запись единиц; запись нулей не действует

Следующие управляющие разряды являются общими для большинства устройств:

- 24 × Незанятость: Считывание/установить триггер "незанятости"
- 25 × Пуск: Считывание/установить триггер пуска
- 26 W Отключение сигнального триггера: сбросить сигнальный триггер
- 27 W Занятость: сбросить триггер занятости
- 28 W Останов: сбросить триггер пуска

Стандартный цикл для чтения знаков в устройстве TR5 следующий:

- I. Знак считывается автоматически в информационные триггеры и устанавливается сигнальный триггер
- II. Знак передается (при помощи программы в постоянной памяти) из информационных триггеров
- III. Сигнальный триггер гасится
- IV. Возврат в (I), если должно считываться большее количество знаков
- V. Установить триггер останова

13.6.2. АДРЕСАЦИЯ ЗУ V

Каждое устройство связано с одним или более регистрами памяти V. Устройства того же типа расположены в последовательных регистрах (или группах последовательных регистров); используются только старшие, наиболее значащие половины слов.

Адреса внешних устройств в памяти V имеют вид:

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
I	I	0	0	0	0	0	0	0	I	0	0	0	δ	δ	δ	δ							0	0	0
память								запасные					внешн. устр.	запа- сные	тип			номер							
V																									

"Типы" внешних устройств и соответствующие разряды прерывания в регистре 2 для памяти V Центральной вычислительной машины следующие:

Тип	0	Устройство считывания с перфокарт 0-3 (P2)
Тип	1	Ксерографические печатающие устройства (P2)
Тип	2	TR7 устройства, считывающие с бумажных лент 0-3 (P2)
Тип	3	Графические выводные устройства 0-I (P2)
Тип	4	Печатающие устройства 0-I "ICT" (P2)
Тип	5	
Тип	6	Быстродействующие ленточные перфораторы 0-3 (P3)
Тип	7	TR5 устройства считывания с бумажных лент 0-II (P3)
Тип	8	Телетайпы 0-II (P3)
Тип	9	Перфораторы 0-I (P4)
Тип	10	Флексорайтеры 0-I5 (если нужны) (P4)
Тип	11	Телепринтеры 0-I5 (P4)
Тип	12	Синхронизатор 0-I и звуковой сигнализатор (P4)
Тип	13	
Тип	14	Сигнальные устройства

13.6.3. РАЗЯДЫ ЗУ V

Тип 0. Устройства, считывающие с перфокарт (максимум 4)

Регистр	Разряд		
0-3	47-36	R	Информация
		W	Останов
		W	Занятость
		W	Выключение сигнального разряда
		×	Пуск
8-II	47-36	×	Незанятость
		×	Контрольная информация (номер считываемого устройства 0-3 соответственно)
		×	
		×	

Тип I. Ксерографические печатающие устройства (максимум 2)

Регистр	Разряд		
0-I	28	W	Останов
		W	Занятость
		W	Выключение сигнального разряда
		×	Пуск
		×	Незанятость
8-9	47-24	W	Информация. Четыре шестизначных знака (номер ксерографического печатающего устройства 0-I соответственно)
		W	

Тип 2. TR7 устройства считывания
с бумажных лент
(максимум 4)

Регистр	Разряд	
0-3	33 R	Тумблер 5/7 переключатель ("1" если лента с 5 дорожками; "0" если лента с 7 дорожками)
	32-29 R	Информация. Разряды 31 и 32 равны нулю для ленты с 5 дорожками
	28 W	Останов
	27 W	Занятость
	26 W	Выключение сигнального разряда
	25 X	Пуск
	24 X	Незанятость
8-II	26 W	Контрольное выключение сигнального разряда
	32-26 R	Контроль информации. Информация считывается в устройстве контрольного чтения (для считывающих устройств № 0-3 соответственно)

Тип 3. Графикостроитель
(максимум 2с)

Регистр	Разряд	
0-I	30 W	Открыть затвор (shutter)
	29 W	Закрыть затвор (shutter)
	28 W	Останов
	27 W	Занятость
	26 W	Отключить сигнальный разряд
	25 X	Пуск
	24 X	Незанятость
8-9	47-36 W	x-координата (номер графического вывода 0-I соответственно)
	35-24 W	y-координата

Тип 4. Печатающее устройство "ICT"
(максимум 2)

Регистр	Разряд	
0-I	36 W	Сигнал останова лентопротяжки. При печати большого массива чисел,

35	W	бумага до останова проходит еще 5 строчек (позиций) Шаговая подача бумаги. Бумага помещается в следующую позицию, определяемую перфорацией
34	W	Подать на 5 строк
33	W	Подать на 4 строки
32	W	Подать на 3 строки
31	W	Подать на 2 строки
30	W	Подать на 1 строку
29	*	При записи - Длительная подача бумаги. Для подачи более чем на пять строк. Вычислительная машина считает строки и дает сигнал останова - протяжки (разряд 36). При чтении - разряд означает занятость механизма подачи бумаги. Читается 1, когда бумага в движении, и 0, когда она останавливается и печатание знаков можно возобновить.
28	W	Остановиться. Гасится триггер пуска и выключается мотор
27	*	При записи - Занятость устройства При чтении - Отметка. Этот сигнал появляется один раз за каждый оборот печатающего барабана, когда выбирается знак - отметка.
26	*	Запись - Выключить сигнальный разряд. Чтение - Кончается бумага. Этот сигнал появляется, когда остается около двух футов бумаги. Вычислительное устройство остановит печатающее устройство в удобной точке, например, в конце строки
	25 X	Пуск
	24 X	Незанятость
2-3	47-24 W	Информация. Позиции знака от 0 до 23 (т.е. левая сторона) печатающих устройств от 0 до I соответственно
4-5	47-24 W	Информация. Позиции знака от 24 до 47 печатающих устройств от 0 до I соответственно
6-7	47-24 W	Информация. Позиции знаков 48-71 печатающих устройств 0-I соответственно
8-9	47-24 W	Информация. Позиции знака 72-95 печатающих устройств 0-I соответственно
10-II	47-24 W	Информация. Позиции знаков 96-II9 печатающих устройств 0-I соответственно

Тип 6. Быстродействующие ленточные
перфораторы
(максимум 4)

Регистр	Разряд	
0-3	35-29	W Информация. Пробивается отверстие между разрядами 31 и 32
	28	W Останов
	27	W Занятость
	26	W Выключение сигнального разряда
	25	x Пуск
	24	x Незанятость

Тип 7. Устройства считывания с
бумажных лент "TR5"
(максимум 12)

Регистр	Разряд	
0-II	33	R Тумблер 5/7 ("I", если лента с 5 дорожками, "0", если с 7-ью)
	32-29	R Информация. Отверстие пробивается между разрядами 28 и 29. В разрядах 31 и 32 записываются нули для ленты с 5-ью дорожками
	28	W Останов
	27	W Занятость
	26	W Выключение сигнального разряда
	25	* Пуск
	24	* Незанятость

Тип 8. Перфораторы - телетайпы
(максимум 12)

Регистр	Разряд	
0-II	35-29	W Информация. Отверстие пробивается между разрядами 31 и 32
	28	W Останов
	27	W Занятость
	26	W Выключение сигнального разряда
	25	x Пуск
	24	x Незанятость

Тип 9. Перфораторы для перфокарт
(максимум 2)

Регистр	Разряд	
0-I	47-40	x Информация. Один разряд каждого знака от 0 до 7, где есть знак 0, является первым в устройстве для чтения с перфокарт. При записи - Информация, которую надо затем пробить. При чтении - Контрольная информация о считывании считывается с предыдущей карты
	29	W Смещение. Карточку можно поместить в приемник, например, когда контрольная информация о считывании неправильна
	28	W Останов
	27	W Занятость
	26	W Выключение сигнального разряда
	25	* Пуск
	24	* Незанятость
2-3	47-24	* Информация (информационные регистры 0-I). Один разряд каждого знака от 8 до 31, 0 или I перфораторов, соответственно
4-5	47-24	* Информация (см. информационные регистры 0-I). Один разряд каждого знака от 32 до 55, от 0 или I перфораторов соответственно
6-7	47-24	* Информация (информационные регистры 0-I). Один разряд каждого знака от 56 до 79, 0 или I перфораторов соответственно

Тип 10. Печатающие устройства
"Флексорайтеры"

(максимум 16, если необходимо)

Регистр	Разряд	
0-I5	37	W Пуск перфоратора
	36	W Останов перфоратора
		Они позволяют вычислительной маши-

			не пробивать и печатать или только печатать
35-29	W	Информация. Записать знак, который надо напечатать	
28	W	Останов	
27	W	Занятость ввода	
26	W	Выключение сигнального разряда	
33	R	Незанятость выхода	
		При "I" вычислительная машина может печатать, когда:	
		а) Тумблер ручной/автоматический на флексорайтере находится в положении "автомат"	
		в) Флексорайтер свободен для ввода	
32-26	R	Информация. Прочсть напечатанные знаки	
25	*	Пуск	
24	*	Незанятость ввода. Когда ввод свободен, читается "0" (незанято), а пуск устанавливает I (начато), световой индикатор на флексорайтере дает оператору возможность начать печатать	

Тип II. Телепринтеры
(максимум I6)

Регистр	Разряд	
0-15	35-29	W Информация
	28	W Останов
	27	W Занятость
	26	W Выключение сигнального разряда
	25	* Пуск
	24	* Незанятость

Тип I2. Синхронизатор и звуковой сигнализатор

а) Счетчик команд

Регистр	Разряд	
0	46-36	R Информация. I добавляется к 36 разряду, каждый раз, когда выполняется команда, за исключением случаев, когда: а) управление работает в режиме прерывания

		б) остановлен синхронизатор
28	W	Останов
26	W	Выключение сигнального разряда
25	*	Пуск

(б) Синхронизатор

I	43-24	R	Гринвичское среднее время в секундах синхронизатора, имеющего счетчик времени на 7 дней, который не останавливается, когда машина выключена
	26	W	Выключение сигнального разряда

(в) Звуковой сигнализатор

8	25	W	Пуск. Производится один щелчок в динамике сигнализатора
---	----	---	---

Тип I4. Сигнальные разряды

Регистр	Разряд	
0	24	R Счетчик команд
	25	R Синхронизатор
I	24-3I	R Телепринтеры 0-7 соответственно
2	24-3I	R Телепринтеры 8-15 соответственно
3	24-3I	R Флексорайтеры 0-7
4	24-3I	R Флексорайтеры 8-15
6	24-25	R Перфораторы 0-I
8	24-3I	R Телетайпы 0-7
9	24-27	R Телетайпы 8-II
II	24-3I	R TR - устройства считывающие с перфоленты 0-7 соответственно
I2	24-27	R То же 8-II
I4	24-27	R Быстродействующие перфораторы 0-3
I6	24-25	R Печатающие устройства ст - 0-I соответственно
I8	24-25	R Графикостроители 0-I
I9	24-27	R Контрольщики TR7(check's)0-3
20	24-27	R TR7
22	24-25	R Ксерографические печатающие устройства 0-I
24	24-27	R Устройства считывания с перфокарт 0-3

Логическая операция "ИЛИ" разрядов регистра 24 является сигналом PI памяти у для Центральной вычислительной машины.

(4) M/E - фиксируются в E

в) Экстракоды для В-регистров (однократная В-модификация). (возможны 256 кодов операций).

Если $f_9 = 1$, $f_8 = 0$, M/E=M и I/ME=ME, то

- (1) $n + bm \rightarrow BII9$
- (2) M + "I" \rightarrow M как обычно
- (3) f E \rightarrow так же, как выше
- (4) M/E \rightarrow E
- (5) адрес ba \rightarrow B I2I

Примечание: Если M/E=E, но сохраняются другие условия, то пункт (2) опускается.

(II) Выход из экстракода

Если M/E=E, I/ME=ME (т.е. управление экстракодом) и $f_9 f_8 f_7 f_6$ есть

0 I 0 I в В коды

или

0 I I I А коды,

то выполняется соответствующий основной код ($f_8 = 0$) и M/E устанавливается в M.

14.3. ПЕРЕКЛЮЧЕНИЕ УПРАВЛЕНИЯ В РЕЗУЛЬТАТЕ ПЕРЕРВАНИЯ

14.3.1. ТРИГГЕРЫ ПЕРЕРВАНИЯ

Это набор триггеров, которые расположены в различных частях памяти V. Ниже приводится список этих триггеров и причины, по которым они могут быть установлены. Их адреса даются в разделе I3.

I. Обнаружение ошибки схемами контроля по четности

Если ошибка по четности обнаружена в ЗУ на сердечниках в рабочей или вспомогательной памяти или во время обмена с барабаном или магнитной лентой.

II. Сигнальные разряды

Когда устанавливается один из сигнальных разрядов, связанный с внешним оборудованием (см. также раздел

Б.1.4).

III. Неэквивалентность

Когда обращаются к адресу в главной памяти и дается сигнал "неэквивалентность со всеми регистрами адреса страницы".

IV. Барабаны

Когда завершается обмен с барабаном или когда появляется ошибка при обмене с барабаном (отличная от ошибки по четности). Эта последняя может быть или когда обмен не начался, или не закончился некоторое время спустя после задания команды обмена.

V. Неисправность магнитной ленты

Когда встречается неисправность (например, понижение вакуума), отличная от ошибки по контрольной сумме.

VI. Переполнение порядка чисел (EO)

Когда порядок сумматора превышает +127 после арифметической операции.

VII. Обращение к заблокированной странице (LO)

Когда производится обращение к адресу на странице и эта страница имеет включенный разряд блокировки и I/ME=ME. Команда не выполняется.

VIII. Запрещенное обращение (sv)

Когда
(а) производится обращение к одному из запрещенных ЗУ (например, памяти V и вспомогательной памяти) и или (1) используется главное управление, т.е. M/E=M и I/ME=ME
или (2) предполагаемый адрес в случае операнда
или (в) предпринимается попытка записать в постоянную память
Во всех случаях команда не выполняется.

Примечание: SA/PA триггер устанавливается при не-эквивалентности, блокировки и запрещенном обращении. Это указывает, когда произошло прерывание при команде или операнде.

14.3.2. РАБОТА СИСТЕМ УПРАВЛЕНИЯ

ПРИ ПРЕРЫВАНИИ

Если установились один или несколько триггеров прерывания (и прерывание не запрещено), то перед выполнением команды производятся следующие действия:

(1) "0" → I

Там, где "0", есть адрес 2048 в постоянной памяти (1024 для pilot model).

(2) I/ME устанавливается в I

Прерывания запрещаются, когда

(а) I/ME = I или

(в) установлен триггер запрещенного прерывания (например, когда последовательность команд в программе должна выполняться без прерывания)

Для удобства выполнения программы прерывания один из В-регистров (В123) выполнен специальным образом. Он устанавливается нормально, а выдаваемый с него код представляет собой умноженный на 8 логарифм входного кода (по восьми младшим разрядам) в соответствии с таблицей.

Вход								Выдача				
Разряды 23-8	7	6	5	4	3	2	1 0	23-7	6	5	4	3
	0	0	0	0	0	0	0 I	0	0	0	0	0
	0	0	0	0	0	0	I	0	0	0	0	I
	0	0	0	0	0	0	I	0	0	0	I	0
	0	0	0	0	I			0	0	0	I	I
	0	0	0	I				0	0	I	0	0
	0	0	I					0	0	I	0	I
	0	I						0	0	I	I	0
	I							0	0	I	I	I
	0	0	0	0	0	0	0	0	0	0	0	0

Для определения причины прерывания производятся следующие действия в программе прерывания:

- 1) Передача содержимого регистра 2 (из памяти V Центральной вычислительной машины) в регистр В-123
- 2) Передача управления одной из восьми программ

посредством таблицы переходов), модифицированных В-123.

Это определяет тип прерывания и также обеспечивает первоочередность выполнения прерывания.

Если прерывание вызвано сигналами схем контроля на четность, то содержимое регистра 0 (памяти V Центральной вычислительной машины) передается в В-123, и управление (опять посредством таблицы переходов и модификации при помощи В-123) передается соответствующей программе.

Если прерывание произошло по сигнальному разряду внешних устройств, то устройство, вызвавшее прерывание, определяется одним или двумя дальнейшими обращениями в В-123. При прерывании по сигналу устройства считывания с перфокарт, это устройство можно определить одним дальнейшим обращением в В-123. При прерывании по сигналу адреса блока магнитной ленты, соответствующий канал можно определить дальнейшим использованием В-123, а сам магнитофон на этом канале известен в программе управления магнитной лентой. Для всех других внешних устройств необходимо передать соответствующее содержимое регистра (31, 30 или 29) "типа" I4" памяти для внешних устройств В-123 и передать управление (модифицирование посредством В-123) к соответствующей программе. Это определяет тип устройства, а дальнейшее использование В-123 определяет устройство, вызвавшее прерывание.

Если прерывание произошло в результате операции "неэквивалентность", то начинается работа программы обмена с барабаном.

Для других прерываний содержимое регистра I памяти V Центральной вычислительной машины передается в В123 и вводится соответствующая программа (например, при переполнении порядка).

Прерывание появляется в результате операции очередности действий и после этого прерывания триггеры, которые привели к прерыванию, гасятся (например, сигнальный разряд в устройстве, считывания с перфокарт). Если дальнейшее прерывание было запрещено во время выполнения программы прерывания, то второе прерывание разрешается, когда триггер запрещения прерывания гасится при выходе из программы первого прерывания.

14.4. ТРИГГЕР "ПАРЫ"

Он устанавливается, когда четная команда должна быть считана из главной памяти, а M/E=M и I/ME=ME.

Он гасится:

- (1) когда "четная" команда должна считываться из главной памяти и M/E=E или I/ME=I или
 - (2) когда команда записывается в M главную память.
- Когда "четная" команда считывается из главной памяти, следующая ("нечетная") команда также считывается в регистр команды. Когда триггер "пары" включен и требуется

считывание из главной памяти, нечетной команды, то эта команда берется из регистра команды: если триггер "пары" не включен, эта команда считывается из главной памяти.

Команды считываются по одной из рабочей и постоянной памяти, в то время, как триггер "пары" и "нечетная" команда в регистре остаются неизменными.

СО Д Е Р Ж А Н И Е

Раздел 1.	Общее описание	3
I.1.	Структура слова	3
I.2.	Центральное ЗУ	4
I.3.	Обозначение типа ЗУ	7
Раздел 2.	Код операций	9
2.1.1.	Разряды кода операций	9
2.1.2.	Экстракод	10
2.1.3.	Регистры управления и триггер пары	14
2.2.2.	Специальные В-регистры	16
2.2.3.	Разряд В-переноса	17
2.3.1.	Перечень операций	18
2.3.2.	Перечень основных тестовых операций	19
2.3.3.	Перечень основных операций накапливающего сумматора	20
2.4.	Подпрограмма экстракода	21
2.4.1.	Распределение номеров операций	21
2.4.2.	Арифметические экстракоды	21
2.4.3.	Организирующие экстракоды	29
Раздел 3.	Подробное описание кодов основных операций	31
3.1.	Арифметические операции с плавающей запятой	31
3.1.1.	Накапливающий сумматор с плавающей запятой	31
3.1.2.	Операции с плавающей запятой	36
3.2.	Переполнение и выход из разрядной сетки в сторону младших разрядов	40
3.3.	Основные В-операции	41
3.4.	Основные тестовые операции	44
3.5.	Основные операции накапливающего сумматора	46
3.6.	Организирующая программа	52
3.6.1.	Координирование подпрограмм	52
3.6.2.	Структура запоминающего устройства	61
3.6.3.	Программа управления магнитной лентой	66
3.6.4.	Внешние устройства	71
3.6.5.	Работа организующей программы при вводе и выводе	75
Раздел 4.	Главное запоминающее устройство и схема обращения к барабану	83
4.1.	Общие принципы ЗУ одного уровня	83
4.2.	Самообучающаяся программа обращения к барабану	85
4.3.	Команды экстракода обращения к барабану	86
4.4.	Основные обращения к барабану	87
Раздел 5.	Описание внешних устройств	90

5.1.	Структура внешних устройств	90
5.1.1.	Организирующая программа	90
5.1.2.	Вывод по требованию оператора	93
5.1.3.	Типы устройств	94
5.1.4.	Сигнальные триггеры	94
5.2.	Магнитные ленты	95
5.2.1.	Устройство Ампекс	95
5.2.2.	Команды экстракода обращения к магнитным лентам	98
5.2.3.	Очередность обращения к магнитным лентам	104
5.2.4.	Программа поиска адресов блоков	104
5.2.5.	Основные операции с магнитными лентами	105
5.2.6.	Контрольное суммирование при обращении к магнитным лентам	109
5.2.7.	Регистры адресов блоков магнитных лент	110
5.2.8.	Адресация и переадресация магнитных лент	111
5.2.9.	Считывание магнитных лент типа Орион на машине Атлас	112
Раздел 6.	Входной язык	114
6.1.	Стандартная компилирующая программа	114
6.2.	Промежуточная комплектующая подпрограмма	120
Раздел 7.	Коды бумажной ленты, перфокарт и печатающего устройства	124
7.1.	Код бумажной ленты	124
7.1.1.	Код пятидорожечной бумажной ленты	124
7.1.2.	Код семидорожечной бумажной ленты	125
Раздел 9.	Тестовые программы	127
9.1.	Тесты для Центральной вычислительной машины	127
Раздел II.	Описание вычислительной машины (MUSE)	131
II.1.	Структура машины	131
Раздел I3.	Запоминающее устройство "v"	132
I3.1.	Адресация ЗУ "v"	132
I3.2.	ЗУ v Центральной вычислительной машины	132
I3.3.	ЗУ v на магнитных сердечниках	134
I3.4.	ЗУ v на барабанах	135
I3.5.	ЗУ v на магнитных лентах	135
I3.6.	Внешнее ЗУ v	137
I3.6.1.	Управляющие разряды и разряды информации	137
I3.6.2.	Адресация ЗУ v	138
I3.6.3.	Разряды ЗУ v	139
Раздел I4.	Описание систем управления	147
I4.1.	Регистры управления	147
I4.2.	Переключение управления на подпрограммы экстракода	147
I4.3.	Переключение управления в результате прерывания	148

I4.3.1.	Триггеры прерывания	I48
I4.3.2.	Работа систем управления при прерывании	I50
I4.4.	Триггер "пары"	I52

6P-105920

09	1.62 г.
Акт № 2/60	

Сдано в набор 12/УП-1962 г.

Зак.160а

Тир.300

ИТМ и ВТ АН СССР. Москва, Ленинский проспект, 51