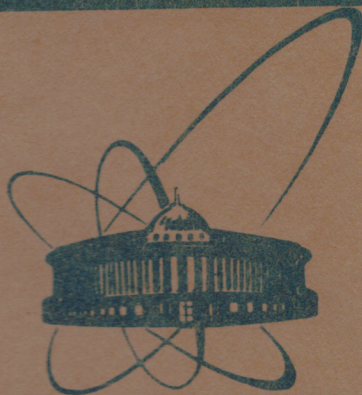


o!



сообщения
объединенного
института
ядерных
исследований
дубна

5320 / 2-80

3/41-80

P10-80-490

А.И.Островной, И.М.Саламатин

СТАНДАРТНЫЕ ПРОГРАММЫ
В СИСТЕМЕ САНПО
ДЛЯ МИНИ-ЭВМ ТИПА СМ-3.

Часть 1

1980

P10-80-490

Островной А.И., Саламатин И.М.

Стандартные программы в системе САНПО
для мини-ЭВМ типа СМ-3.

Часть I

Описан стандарт, принятый для написания программных модулей в системе САНПО. Эти модули являются основной составляющей компонентой при генерации прикладных систем автоматизации экспериментов. Модули реализуют отдельные операции системы и могут быть инициированы как в автоматическом, так и в интерактивном режимах. Описан минимальный интерфейс, позволяющий программировать модули на макроассемблере и ФОРТРАНе.

Работа выполнена в Лаборатории нейтронной физики ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1980

P10-80-490

Ostrovnoj A.I., Salamatin I.M.

Standard Programs in SANPO System
for SM-3 Type Minicomputers.

1. ВВЕДЕНИЕ

При создании программного обеспечения /ПО/ систем автоматизации физического эксперимента большое внимание уделяют его модульной организации. Разработаны различные способы генерации ПО таких систем из набора модулей^{/1-3/}. В системе САНПО /система автоматического накопления и предварительной обработки экспериментальной информации/, предназначенной для работы на ЭВМ типа СМ-3^{/4/}, реализован метод генерации программных систем для конкретных экспериментов из набора модулей^{/5/}. Модуль интерпретируется как программная единица, реализующая одну неделимую операцию системы.

Данная работа состоит из двух частей и посвящена изложению правил написания и оформления программных модулей. В САНПО такие модули названы стандартными программами /СП/. В первой части представлены общие для всех СП организация и правила их программирования, способ доступа простейших СП к данным и сервис системы САНПО, облегчающий реализацию в СП алгоритмов преобразования информации, поступающей в режиме работы ЭВМ на линии с экспериментальной установкой.

2. ОБЩИЕ СВЕДЕНИЯ О СТАНДАРТНЫХ ПРОГРАММАХ СИСТЕМЫ САНПО

Стандартные программы представляют собой программные модули в двоичном перемещаемом формате загрузки REL^{/6/}. Они оформляются в виде подпрограмм типа SUBROUTINE с учетом дополнительных требований, изложенных в данной работе. Программирование СП выполняется с помощью обычных средств, включенных в операционную систему RT-11^{/7/}. СП составляется редактором связей LINK^{/7/} из одной или нескольких подпрограмм, написанных на языках FORTRAN-IV^{/8/} и MACRO-11^{/7,9/}.

Неотъемлемой частью тела СП является документация. Без нее СП не может быть записана в библиотеку^{/10/}, где хранятся все СП, включаемые в прикладные системы. СП используются для компоновки прикладных систем, а также для реализации операторов языка^{/5/}, на котором выполняется описание прикладных систем. Одни и те же СП могут быть использованы в различных прикладных системах, при этом не требуется какой-либо модификации программ или их повторной трансляции. Любая СП может иницииро-

ваться как в автоматическом режиме, в соответствии с программой эксперимента, так и в интерактивном, по приказу с клавиатуры телетайпа.

Все СП в системе имеют одинаковую структуру и равные возможности в отношении доступа к данным, способа работы с ними и взаимодействия с монитором САНПО. Монитор предоставляет СП интерфейс в виде таблиц и системного счетчика, которыми может воспользоваться любая СП. Интерфейс, предоставляемый СП, делится на уровни по сложности работы с ним и степени вмешательства СП в работу системы в целом, т.е. объему действий СП, не поддающихся централизованному контролю их корректности со стороны монитора. В соответствии с применяемым интерфейсом из всех возможных СП выделяются простейшие, которые используют минимальный интерфейс, не требующий знания системы САНПО, таблиц монитора и оперативной базы данных. К такого рода СП относятся прежде всего СП, написанные на языке FORTRAN-IV.

В данном сообщении представлены сведения, необходимые для программирования СП, использующих любой уровень интерфейса, но достаточными эти сведения являются только для программирования простейших СП, которые используют вышеупомянутый минимальный интерфейс.

3. СТРУКТУРА СП И ДОСТУП К ДАННЫМ

СП, как уже сказано выше, является обычной подпрограммой типа SUBROUTINE, полученной с помощью стандартных средств операционной системы RT-11. Управление передается СП командой /7/ вида

JSR PC, <адрес старта СП>.

Последней исполняемой командой СП должна быть команда RTS PC. На языке FORTRAN-IV /8/ эти команды соответствуют операторам CALL <имя подпрограммы> (список аргументов) и RETURN.

Доступ к данным СП получает с помощью интерфейсной таблицы /ИТ/. В момент инициирования СП пятый регистр /R5/ содержит

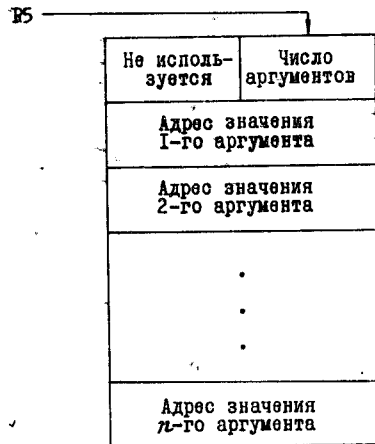
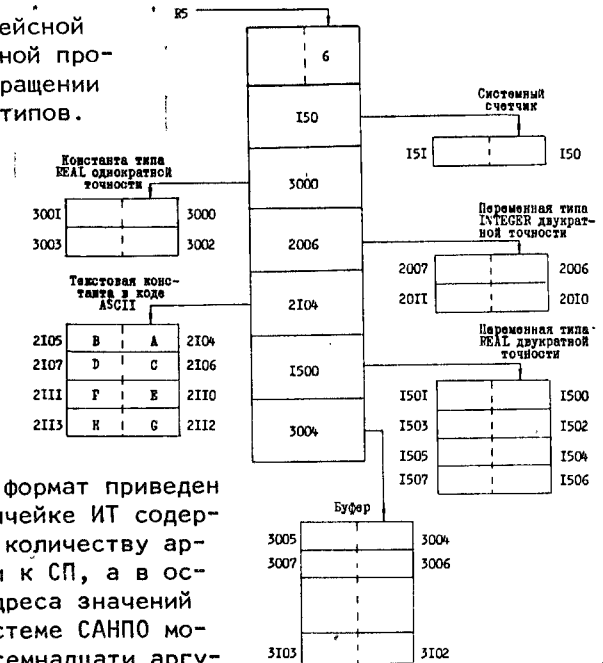


Рис.1. Формат интерфейсной таблицы.

Рис.2. Пример интерфейсной таблицы для стандартной программы, имеющей в обращении аргументы различных типов.



адрес начала ИТ. Ее формат приведен на рис.1. В первой ячейке ИТ содержится число, равное количеству аргументов в обращении к СП, а в остальных ячейках - адреса значений аргументов. СП в системе САНПО могут иметь не более семнадцати аргументов, причем первый из них - это всегда системный счетчик, значение которого определяется системой. Он занимает одно слово оперативной памяти и является системной переменной типа INTEGER. Его назначение и способ использования описаны в следующем параграфе. Остальные же шестнадцать аргументов определяет программист, составляющий данную СП.

В качестве аргументов прикладных СП могут быть использованы переменные, буфера и константы. Переменные и буфера содержат числа типа INTEGER или REAL однократной или двойной точности. Форматы чисел приведены в приложении и совпадают со стандартом, принятым в языке FORTRAN-IV для ЭВМ PDP-II /9/. Константы возможны числовые и текстовые. Числовые константы являются числами уже упомянутых типов. Их отличие от переменных состоит в том, что они не изменяются стандартной программой во время ее работы. Текстовые константы могут быть представлены в кодах ASCII и RADIX -50 /7/. Текстовая константа в коде ASCII должна содержать не более восьми символов, а в коде RADIX -50 - не более двенадцати.

На рис.2 приведен частный пример ИТ для СП, имеющей в списке аргументов обращения переменные, константы и один буфер. На рисунке указаны абсолютные адреса байтов оперативной памяти, где хранятся значения аргументов. В позициях ИТ для

каждого из аргументов стоят адреса младших байтов, начиная с которых значения аргументов располагаются в оперативной памяти.

4. СЕРВИС СИСТЕМЫ

Сервис системы состоит в том, что СП предоставляется счетчик, значение которого вырабатывает монитор САНПО. Использование сервиса для СП и заключается в применении ею этого счетчика. На структуру СП, использующих сервис системы, наложен ряд дополнительных ограничений. Для таких СП фиксировано назначение первых трех аргументов. Первым остается счетчик, а вторым и третьим являются буфера. Считается, что всякая СП, использующая сервис системы, выполняет преобразование исходной информации из буфера-источника /второй аргумент/ и результат заносит в буфер-приемник /третий аргумент/. Преобразование выполняется фиксированными квантами информации, например, n_1 байтов источника стандартной программой будет преобразовано в n_2 байтов приемника. Назовем такое действие тактом преобразования.

Счетчик, предоставляемый СП, определяет для нее объем работы в виде количества тактов преобразования, которые она должна выполнить. Эта СП по существу должна реализовать цикл из тактов преобразования /для пользователя СП - неявный цикл/. Длина цикла задана значением счетчика.

Можно считать, что для СП, использующей сервис системы, исходная информация находится в массиве длиной $n_1 \times n_{сч}$, а результаты должны быть представлены в виде массива длиной $n_2 \times n_{сч}$ байтов, где n_1 и n_2 - значения квантов преобразования соответственно для источника и приемника, а $n_{сч}$ - значение счетчика. Начальные адреса этих массивов в момент инициализации СП будут находиться во второй и третьей ячейках ИТ.

После каждого такта преобразования в СП следует уменьшать счетчик на единицу. По окончании цикла счетчик тактов преобразования окажется равным нулю. СП может выполнять любое число тактов преобразования, но не больше заданного. Если СП выполнила меньше тактов, чем было задано, то счетчик после завершения работы СП не равен нулю. В этом случае информация, оставшаяся необработанной, не будет утеряна или испорчена, система обеспечит ее целостность, и она может быть обработана в результате последующих обращений к той же самой или другой СП.

Кванты преобразования определяются программистом во время создания тела СП. Их значения записываются в обязательной

```

SUBROUTINE F1(ICNTER,A,B)
DIMENSION A(1), B(1)
N=ICNTER
DO 10 K=1,N
ICNTER=ICNTER-1
B(K)=A(2*K-1)+A(2*K)
10 CONTINUE

```

Рис. 3. Фрагмент текста стандартной программы, использующей сервис системы, на языке FORTRAN-IV.

части документации СП в качестве одной из ее характеристик. Значения квантов могут варьироваться в пределах от нуля до 255 байтов, если источник и приемник находятся в оперативной памяти. В случае, если оба или один из них находятся на внешних запоминающих устройствах, то значения квантов равны и лежат в пределах от 0 до $2^{16} - 1$. Значения квантов равны нулю, если данная СП не использует сервис системы.

Начальное значение счетчика числа тактов преобразования вычисляется монитором системы перед обращением к СП на основе данных об имеющемся в наличии количестве готовой к обработке информации, объеме памяти для результатов и величине квантов преобразования. Значение счетчика $n_{сч}$ вычисляется по формуле

$$n_{сч} = \begin{cases} N_1/n_1, & \text{если } N_1/n_1 \leq N_2/n_2, \\ N_2/n_2, & \text{если } N_1/n_1 > N_2/n_2. \end{cases}$$

Здесь N_1 и N_2 - длины, а n_1 и n_2 - значения квантов преобразования соответственно для буфера-источника и буфера-приемника.

Такой способ вычисления счетчика позволяет заказать стандартной программе с помощью значения счетчика максимально допустимую по объему операцию. Заметим, что при использовании СП необходимо, чтобы длины буфера-источника и буфера-приемника были кратны значениям квантов преобразования.

Возможны отступления от ограничений, наложенных на структуру и способ функционирования СП, использующей сервис системы. Так, третий аргумент может быть не буфером, а, например, переменной или константой, или вовсе отсутствовать. В этом случае

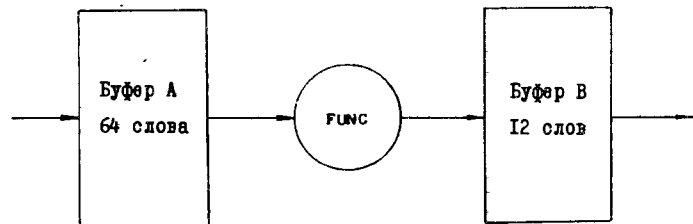


Рис.4. Простейший процесс преобразования информации.

считается, что СП обрабатывает источник квантами в n_1 байтов, а приемника просто не существует. Значение n_2 должно быть равным нулю. Такая СП может осуществлять, например, вывод группами по n_1 байтов оперативной памяти на телетайп, дисплей, передачу по линии связи и т.п. Значение счетчика в момент инициирования СП будет равно N_1/n_1 .

На рис.3 приведен фрагмент СП, использующей сервис системы. Эта программа складывает два соседних числа типа REAL из источника и результат засылает в приемник. Таким образом, эта СП преобразует каждые четыре слова источника в два слова приемника /кванты преобразования равны соответственно $n_1=8$ и $n_2=4$ /.

5. ИСПОЛЬЗОВАНИЕ СЕРВИСА СИСТЕМЫ СП, ВКЛЮЧЕННОЙ В ПРОЦЕСС ОБРАБОТКИ

В данном параграфе мы опишем на примере работу СП, использующей сервис системы и включенной в процесс обработки. Понятие процесса, его полное определение, введено в работе /5/, а здесь мы рассмотрим только фрагмент простейшего процесса /см. рис.4/, который выполняет преобразование исходной информации из буфера А по алгоритму, реализованному программой FUNC, и результат засылает в буфер В.

В примере буфер А имеет длину 64 слова, а буфер В - 12 слов. Стандартная программа FUNC использует сервис системы и осуществляет преобразование каждых 16 слов источника /буфер А/ в 4 слова приемника /буфер В/. Кванты преобразования равны $n_1=32$ и $n_2=8$. СП FUNC иницируется монитором, когда буфер А /источник/ заполнен информацией, готовой к переработке. При первом обращении к FUNC монитор формирует ИТ, состояние которой в этот момент времени приведено на рис.5. Монитор задает значение счетчика, равное 3, и начальные адреса

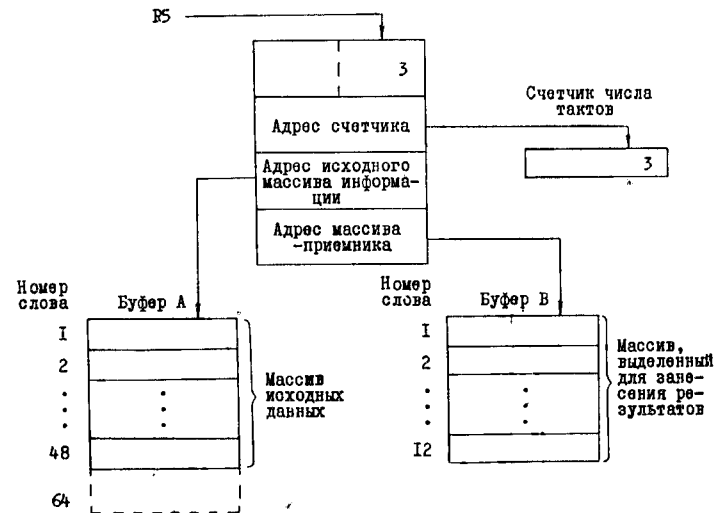


Рис.5. Состояние интерфейсной таблицы и буферов перед первым иницированием программы FUNC.

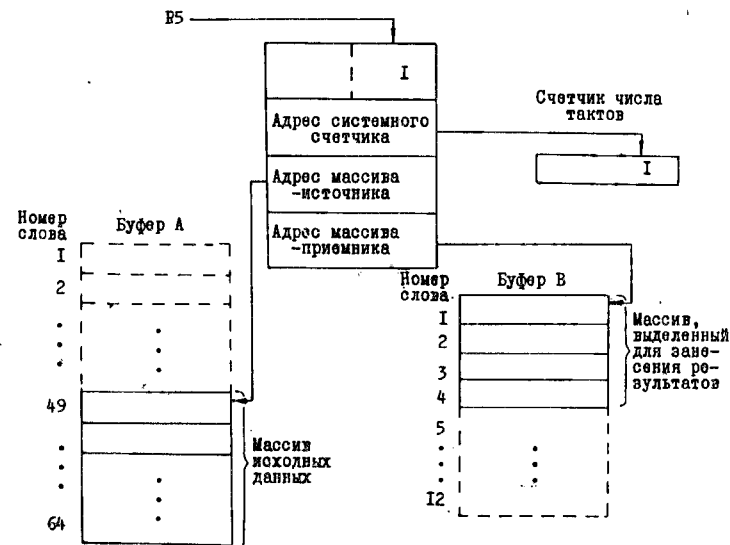


Рис.6. Состояние интерфейсной таблицы и буферов перед вторым иницированием программы FUNC.

используемых массивов. СП получает в виде исходной информации массив длиной в $16 \times 3 = 48$ слов. Остальная часть буфера А игнорируется /на рис.5 она изображена пунктиром/. Для результатов СП имеет массив длиной в $4 \times 3 = 12$ слов /весь буфер В/. По окончании работы СП счетчик числа тактов будет равен нулю, а выделенные для работы участки буферов FUNC переработает в соответствии с заложенным в нее алгоритмом.

После того как результаты работы FUNC /буфер В / будут обработаны другими СП и буфер В будет освобожден от заполнившей его информации, монитор снова иницирует СП FUNC. При этом будет учтено, что первые 48 слов буфера А уже обработаны. Перед вторым иницированием FUNC монитор задаст ИТ, состоящие которой приведено на рис.6. Значение счетчика будет равно 1, адрес массива-источника равен адресу 49-го слова буфера А /начало необработанной области буфера/, а адрес массива приемника будет адресом первого свободного слова буфера В /первого слова, т.к. буфер освобожден полностью/. Таким образом, СП FUNC будет представлен массив исходных данных длиной 16×1 слов в буфере-источнике, а для результатов СП во время работы будет использовать массив длиной 4×1 слова в буфере-приемнике. На рисунке неиспользуемые части буферов изображены пунктиром.

В этот раз FUNC обработает один такт преобразования, после чего монитор определит, что исходная информация исчерпана, объявит буфер А свободным, и FUNC будет иницирована вновь лишь после его заполнения новой порцией данных. Рассмотренный нами фрагмент процесса является по существу частным примером элементарного процесса ^{1/5/} - единицы программирования параллельно исполняемых процессов накопления и предварительной обработки в системе САНПО.

6. РЕКОМЕНДАЦИИ ДЛЯ ПРОГРАММИРОВАНИЯ ПРОСТЕЙШИХ СП

При программировании СП неизбежно возникают вопросы: какими качествами должны обладать СП? Какие алгоритмы удобно реализовать в виде СП, использующих сервис системы? Каким образом разбить решение общей задачи накопления и предварительной обработки на несколько СП? и другие. На эти вопросы в настоящее время достаточно полный ответ дать невозможно. Но тем не менее сейчас можно привести некоторые рекомендации общего характера, которые помогут получить более быстродействующие операции, реализованные в виде СП, расширить область их применения, облегчить выбор алгоритмов, которые следует реализовать в виде одной или нескольких СП, использующих сервис системы.

Для любых СП желательно такое построение алгоритма, при котором:

1. СП в процессе работы не портит себя /все изменения СП выполняет только со своими аргументами либо перед окончанием своей работы восстанавливает все "испорченные" области программы и рабочие ячейки/.

2. СП является обратимой, т.е. она не портит исходных данных. Такую СП можно, например, два раза подряд иницировать /не загружая в ОП повторно/ с одними и теми же аргументами и получить одинаковый результат. Заметим, что запрограммированные таким способом СП обеспечивают более надежную работу прикладной системы.

Приведенные рекомендации не являются обязательными, но факт наличия или отсутствия каждого из этих свойств отражается в документации СП.

Приведем также рекомендации, связанные с использованием сервиса системы:

3. Кванты преобразования следует выбирать как можно меньшими. Потери скорости работы СП не возникнет, но они будут применимы к буферам, имеющим более широкий диапазон размеров.

4. Алгоритмы, инвариантные по отношению к размерам обрабатываемых массивов, следует реализовать в виде СП, использующих сервис системы. Такими алгоритмами являются алгоритмы сортировки, фильтрации, преобразования форматов и т.д.

5. Алгоритмы, не обладающие упомянутой в предыдущем пункте инвариантностью, следует разбивать на части, если это возможно, с тем, чтобы все неоднородности алгоритма локализовать в рамках квантов преобразования. Такие алгоритмы не обладают инвариантностью в пределах одного кванта и обладают ею во всем массиве данных, если обрабатывать его указанными квантами.

6. Реализацию сложных алгоритмов, если возможно, следует разбивать на несколько СП, с учетом предыдущих двух рекомендаций.

7. Возможно такое построение алгоритма СП, при котором выработанный в соответствии с описанным правилом системный счетчик может использоваться СП отличным от предложенного /см.п.4/ способом. СП в зависимости от алгоритма может сразу занести в счетчик нулевое значение или уменьшить его на несколько единиц за один такт преобразования. Формально считается, что СП выполнила заказанную операцию, если после ее работы значение счетчика равно нулю.

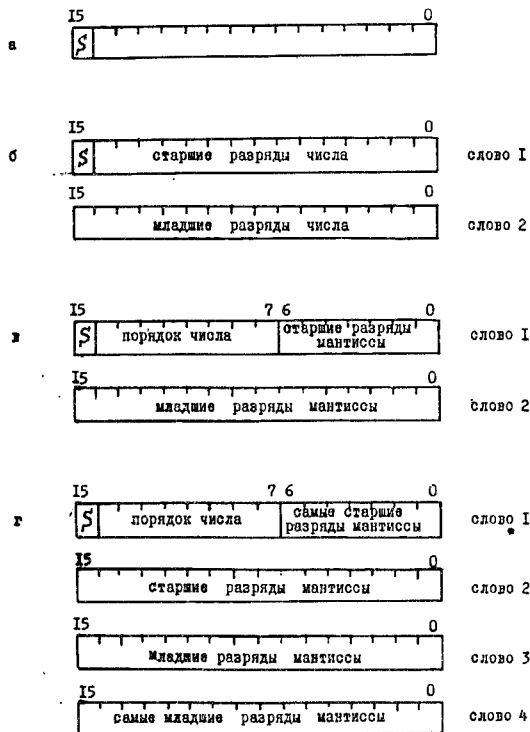


Рис.7. Форматы чисел.

чески в соответствии с заданной программой эксперимента либо в интерактивном режиме по приказу с клавиатуры телетайпа.

Принятый стандарт для программирования СП обеспечивает им следующие качества:

1. Независимость от места расположения и длины обрабатываемых массивов данных. Заметим, что это качество обеспечивается для СП, написанных как на MACRO-II, так и на FORTRAN-IV.
2. Ориентацию на обработку данных, поступающих в реальном масштабе времени. Особенность задач этой области состоит в том, что объем исходных данных, поступающих для обработки, либо неопределен, либо его нежелательно ограничивать.
3. Организацию неявных циклов за счет использования сервиса системы прикладными СП.
4. Возможность программировать в виде одной или нескольких СП достаточно широкий класс алгоритмов.

Использование пакета СП в системе САНПО позволяет ориентировать средства генерации прикладных систем на определенный

7. ЗАКЛЮЧЕНИЕ

Принятый в системе САНПО стандарт программирования СП позволяет создать средства генерации программных систем для конкретных экспериментов, ориентированные на пользователя, не имеющего специальной подготовки в области системного программирования. При описании прикладных систем не требуется знать структуру СП, используемых для компоновки, и детально описывать их взаимосвязи. Одни и те же СП могут быть использованы в различных прикладных системах без какой-либо модификации, что обеспечивает преемственность развития прикладных систем. Во время проведения эксперимента они могут быть инициализированы автоматиче-

класс экспериментов и представить эти СП в виде операций проблемно-ориентированного языка высокого уровня.

Авторы выражают благодарность Г.Я.Яновскому, Ю.Намсраю и Г.Балуке за полезные обсуждения и Т.Б.Журавлевой за оформление рукописи.

ПРИЛОЖЕНИЕ. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В СИСТЕМЕ САНПО

Формат чисел в системе САНПО принят аналогичным разработанной фирмой DEC для языка FORTRAN-IV^{/9/}. Целое число однократной точности занимает одно слово оперативной памяти /рис.7а/. Здесь S обозначает знак числа: S = 1, если число отрицательное, и S = 0, если оно положительное. Целое число двукратной точности занимает два последовательных слова /рис.7б/. Число типа REAL однократной точности занимает два слова оперативной памяти /рис.7в/. Здесь S - знак мантиссы. Порядок числа занимает 8 разрядов и может принимать значения от 000_8 до 377_8 , что соответствует в десятичном представлении интервалу $2^{-128} \div 2^{+127}$. Мантисса имеет длину 24 разряда, но для нее отводится в памяти только 23 разряда. Старший разряд мантиссы в двоичном виде всегда равен единице, т.к. числа типа REAL представлены в нормализованном виде, поэтому эта старшая единица мантиссы всегда подразумевается, но физически не записывается в памяти ЭВМ. Число типа REAL двукратной точности занимает четыре слова /рис.7г/.

ЛИТЕРАТУРА

1. Талныкин Э.А. Модульное программирование в задачах сбора и обработки экспериментальных данных. Автометрия, 1976, № 1.
2. Жук В.И., Шитиков Б.И. Модульное программирование на разговорном языке описания блок-схем. Автометрия, 1976, № 1.
3. Taff L.M., Sporel F. Nucl.Instr. and Methods, v.160, No 1, p.147-158.
4. Наумов Б.Н., Боярченко М.А., Кабалевский А.Н. Управляющий вычислительный комплекс СМ-3. Приборы и системы управления, 1977, № 10.
5. Балука Г. и др. ОИЯИ, P10-12960, Дубна, 1980.
6. RT-11 Software Support Manual (DEC-11-ORUGA-C-D). DEC, Maynard, Massachusetts, 1975, p.342.

7. RT-11 System Reference Manual (DEC-11-ORPGA-B-D). DEC, Maynard, Massachusetts, 1975, p.755.
8. PDP-11 FORTRAN Language Reference Manual (DEC-11-LFLRA-B-D). DEC, Maynard, Massachusetts, 1974, p.173.
9. FORTRAN-IV User's Guide (DEC-11-LMFVA-B-D). DEC, Maynard, Massachusetts, 1975, p.82.
10. Балука Г., Саламатин И.М., Хрыкин А.С. ОИЯИ, 10-12545, Дубна, 1979.

Рукопись поступила в издательский отдел
10 июля 1980 года.