

на 10 ун. 14.06.76

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА



Ц 8405
3-224

18/III-75

10 - 8873

Ю.П.Залаторюс

3039/2-75

ОБ УСКОРЕНИИ ФОРМАТНОГО ВЫВОДА
В СИСТЕМЕ "ДУБНА" НА ЭВМ БЭСМ-6

1975

10 - 8873

Ю.П.Залаторюс

ОБ УСКОРЕНИИ ФОРМАТНОГО ВЫВОДА
В СИСТЕМЕ "ДУБНА" НА ЭВМ БЭСМ-6



Во время решения задачи на ЭВМ часто приходится осуществлять обмен цифровой информацией (ввод данных, вывод результатов на АЦПУ, запись на МЛ, МБ, считывание с МЛ и т.д.).

При использовании мониторной системы "ДУБНА" в программах, написанных на ФОРТРАНе, для этих целей применяются операторы PRINT, READ, WRITE, PUNCH, ENCODE, DECODE. Весь обмен информацией, который происходит во время счета задачи пользователя с помощью этих операторов, осуществляется так называемой системной программой ввода-вывода при счете (в дальнейшем будем называть программой ввода-вывода).

Программу ввода-вывода составляют 7 системных подпрограмм: IOCONT*, BCDBEG*, BCDREAD*, BCDWRITE*, BINREAD*, IOCONTD*, SWRITE.

Во время трансляции задачи пользователя вместо каждого из операторов типа PRINT, READ, WRITE, PUNCH, ENCODE, DECODE генерируется примерно такая последовательность обращений к автокодным подпрограммам:

I. Обращение типа ,CALL, FT* XXI

Например:

,CALL, FT*571 - для оператора PRINT FT, LIST
,CALL, FT*621 - для оператора WRITE(LUN, FT),

LIST и т.д.

Примечание: здесь и дальше X обозначает десятичную цифру;

FT - метка формата;

LUN - логический номер устройства;

LIST - список величин, участвующих в обмене.

2. Обращение типа ,CALL, FT * XX2

Например:

,CALL, FT * 002 - для оператора PRINT FT,LIST,
когда элементы списка LIST-
величины типа INTEGER или
REAL ;

,CALL, FD * 002 - для оператора PRINT FT,LIST,
когда элементы списка LIST
представляют собой величины
типа DOUBLE PRECISION.

3. Обращение типа ,CALL, FT* XX3 для заканчивания обмена

Примечание: Во время счета задачи обращение типа ,CALL, FT * XX2
происходит для каждого слова (если число DP - то двух
слов) участвующего в обмене, а ,CALL, FT* XX1 и
,CALL, FT* XX3 - только один раз.

Все эти подпрограммы FT * XXX являются, по существу, только
точками входа в вышеозначенные системные подпрограммы. Ниже дана
краткая характеристика подпрограмм программы ввода-вывода. (Подроб-
ное описание программы ввода-вывода дано в /5/).

IOSCONT* - используется другими подпрограммами и содержит
ряд рабочих блоков: блок печати информации об ошибках, блоки посим-
вольного считывания текста формата, блоки посимвольного считывания
введенной информации, блоки запрятывания индекс-регистров. В IOSCONT*
содержится также блок констант и блок рабочих ячеек, общих для всех
подпрограмм ввода-вывода.

BCDBEG* - начинает подготовку к форматному обмену, произво-
дит управление обменом и заканчивает его. BCDBEG* содержит блок
обработки и синтаксического контроля формата, блок кодирования вве-
денного числа.(I часть кодирования) и другие .

BCDWRT* - производит форматный вывод по спецификациям:
/ ,NX,WH,EW.d,FW.d,IW,OW,AW, LW и содержит рабочие блоки:
блок декодирования чисел (REAL,INTEGER,DOUBLE PRECISION), блок
записи декодированной информации посимвольно в буфер вывода и другие.

BCDREAD* - производит форматный ввод (числа типа INTEGER
и REAL) и содержит блок обработки спецификаций: /,NX,OW,AW,LW,
блок кодирования введенного числа (по спец. IW, EW.d,
FW.d) и другие.

VINREAD* - производит бесформатный обмен (ввод и вывод) и
содержит блоки для подготовки к обмену, управлению обменом и завер-
шения обмена.

IOSCONTD* - производит форматный обмен величинами двойной
точности (DP) и содержит блоки для кодирования и декодирования по
спецификации DW.d.

SWRITE - производит запись и считывание информации с МЛ.

При решении задач почти всегда используется форматный вывод (выдача результатов на АЦПУ, запись на МЛ), причем время, затрачиваемое на подготовку к выводу, обработке формата, подготовку к декодированию чисел и само декодирование не так уж мало (для 5000 чисел это время варьируется в среднем где-то между 10 и 20 секундами). На что же затрачивается это время? Оказывается, что время для подготовки к выводу и обработки формата – небольшое, а наибольшее количество (в типичном случае свыше 60%) времени расходуется на подготовку к декодированию и само декодирование чисел.

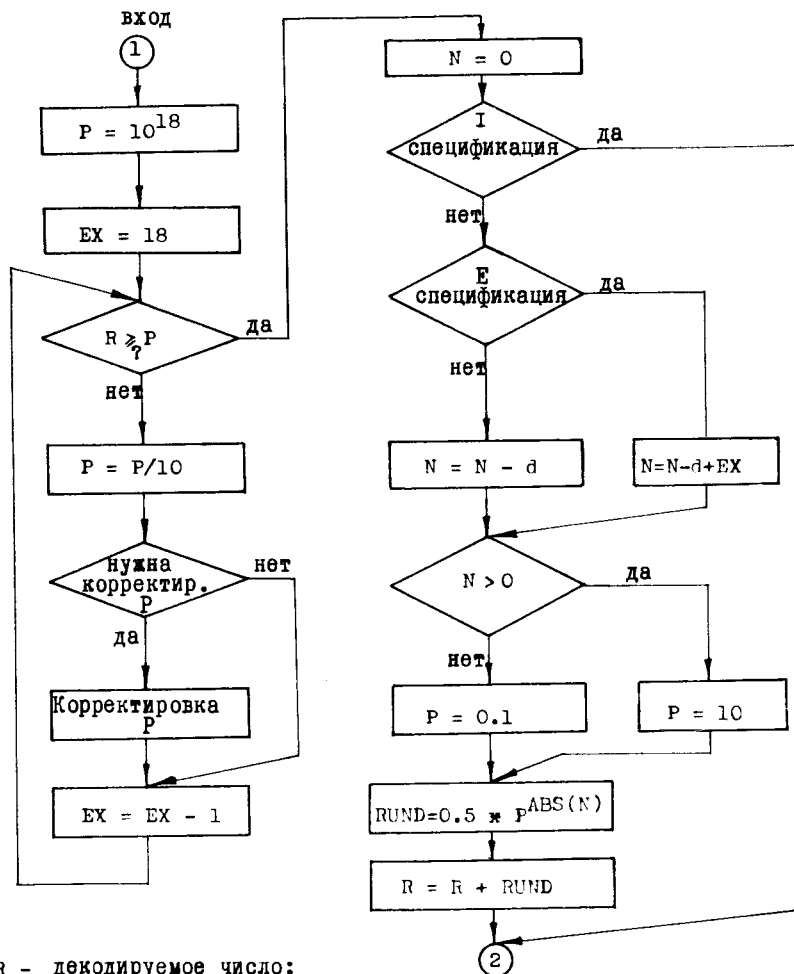
В мониторинной системе "ДУБНА" как и в первом варианте программы ввода-вывода /6/, так и в теперь существующем варианте GPACK /4/ используется алгоритм декодирования чисел (рис. 1а, 2а, 3а).

Хотя самый внутренний цикл (декодирование отдельных цифр целого числа; см. рис. а3) работает быстро, но в целом этот алгоритм содержит ряд недостатков:

1. При вычислении десятичного порядка EX числа R (см. рис. 1а) время вычисления сильно зависит от абсолютной величины числа R . Временные затраты на вычисление EX (для 5000 чисел) даны на рисунке 1в. Имея в виду, что по данному алгоритму в случае спецификации $EW.a$ (наиболее часто используемой) вычисление EX происходит дважды (до и после округления), время, затраченное на вычисление EX , варьируется от 2,2 до 5 сек на 5000 чисел, т.е. от 12 до 28% времени, затрачиваемого для подготовки чисел на вывод.

2. Подготовка значащих десятичных цифр выводимого числа производится порциями по 10 десятичных цифр (рис. 2а, 3а) и часто получается, что подготовлено намного больше цифр, чем нужно (например, если число R , $10^{-1} < R < 10^2$, выводится по спецификации $E 9.2$, то вместо 3 значащих цифр будет подготовлено 11 цифр).

Принципиальная блок-схема подготовки числа R к декодированию и вычисления EX (т.е. десятичного порядка) (блок `WEND` п/л `BCDWRITE*`)



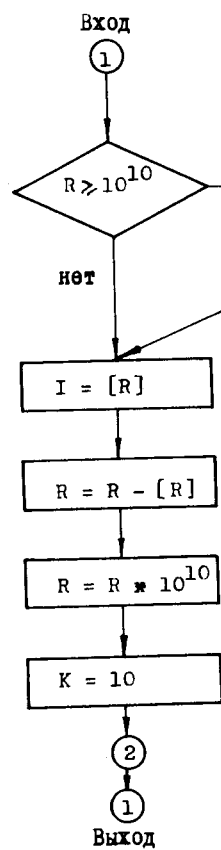
R – декодируемое число;
 d – число выводимых цифр после десятичной точки.

Рис. 1а.

Принципиальная блок-схема подготовки очередной порции

(10 цифр) декодированных цифр

(блок RK* подпрограммы VCDWRIT*)

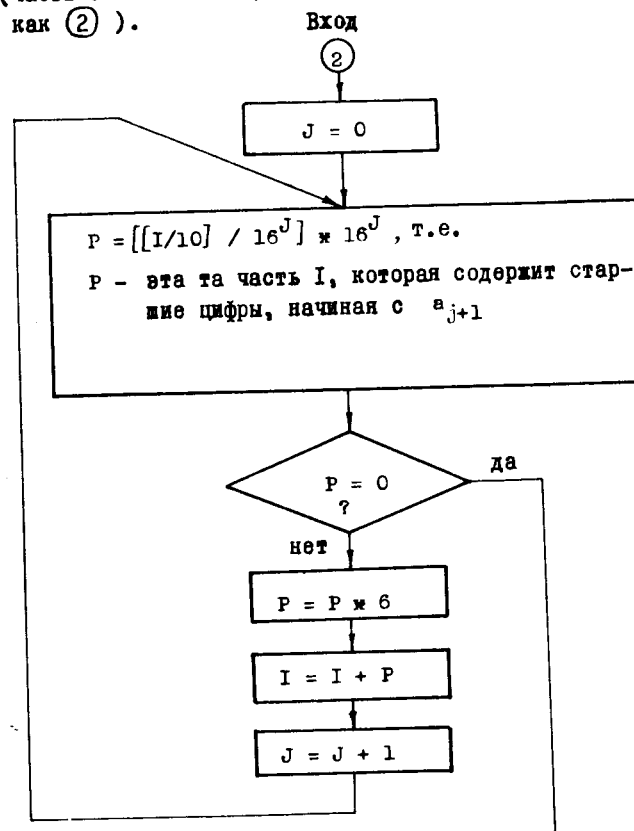


- ② - декодирование целого числа I , $1 \leq I < 10^{10}$ (DIS.АЗ);
- R - декодируемое число (две ячейки) типа INTEGER, REAL или DOUBLE PRECISION;
- [R] - целая часть числа R.

Рис. 2а.

Принципиальная блок-схема декодирования целого числа I , где $10^0 \leq I < 10^{10}$ и $I = a_0 \cdot 10^0 + a_1 \cdot 10^1 + \dots + a_n \cdot 10^n$

(часть блока RK* п/п VCDWRIT* ; в рис. 2а она отмечена как ②).



- ② - выход;
- I - ячейка, в которой находится целое число $1 \leq I < 10^{10}$.

Рис. 3а.

3. При подготовке значащих цифр вначале всегда (за исключением случая $10^9 < R < 10^{10}$) подготавливается к незначащим нулей ($K = 10 - EX$, если $EX \leq 10$ и $K = 20 - EX$, если $EX > 10$, где EX - это десятичный порядок числа R), т.е. если, например, порядок $EX = -5$, то подготавливается $10 - (-5) = 15$ незначащих нулей; если $EX = -11$, то 21 незначащий нуль, на подготовку которых тратится время.

4. Так как сначала подготавливаются ненужные незначащие нули, то при каждом обращении за очередной десятичной цифрой требуется проверить, не нуль ли эта цифра, а если нуль, то значащий или нет, на что опять тратится время.

Поэтому пришлось отказаться от этого алгоритма подготовки чисел к декодированию и декодирования. Алгоритм вычисления десятичного порядка EX числа R был заменен другим (рис. 1с) являющимся, по существу, разновидностью метода деления пополам. Хотя это и потребовало увеличения объема программы на 19 ячеек (из-за таблицы величин 10^{-18} , 10^{-16} , ... 10^{18}), но временной выигрыш оправдал это (рис. 1в).

На вычисление порядка EX (5000 чисел) по новому алгоритму требуется (независимо от абсолютной величины выводимых чисел) около 0.4-0.5 с. Так как по новому алгоритму для спецификации $E^w.d$ вычисление порядка EX происходит однажды, то чистый выигрыш в среднем (на 5000 чисел) составляет около 3,5 с, т.е. около 20% всего времени, затрачиваемого на подготовку чисел к декодированию и само декодирование.

Алгоритм декодирования, ввиду его существенных недостатков, тоже был заменен. Был выбран алгоритм, по которому число R сначала переводится в число R_1 , такое, что: $R_1 = R \cdot 10^{-EX}$ и $1 \leq R_1 < 10$, а само декодирование производится последовательным

выделением целой части R_1 и умножением остатка на 10. Умножение на 10 осуществляется логически, поэтому в сравнении со старым алгоритмом новый алгоритм существенно сокращает время декодирования. Принципиальная блок-схема нового алгоритма дана на рис. д1, сравнительные оценки скорости работы - в таблице 1е.

Автор считает своим долгом выразить благодарность члену-корреспонденту АН СССР Н.Н.Говоруну и сотрудникам ЛВТА В.П.Ширикову и Г.Л.Мазному за внимание и помощь в работе.

Зависимость времени, затрачиваемого на вычисление EX, от абсолютной величины числа R

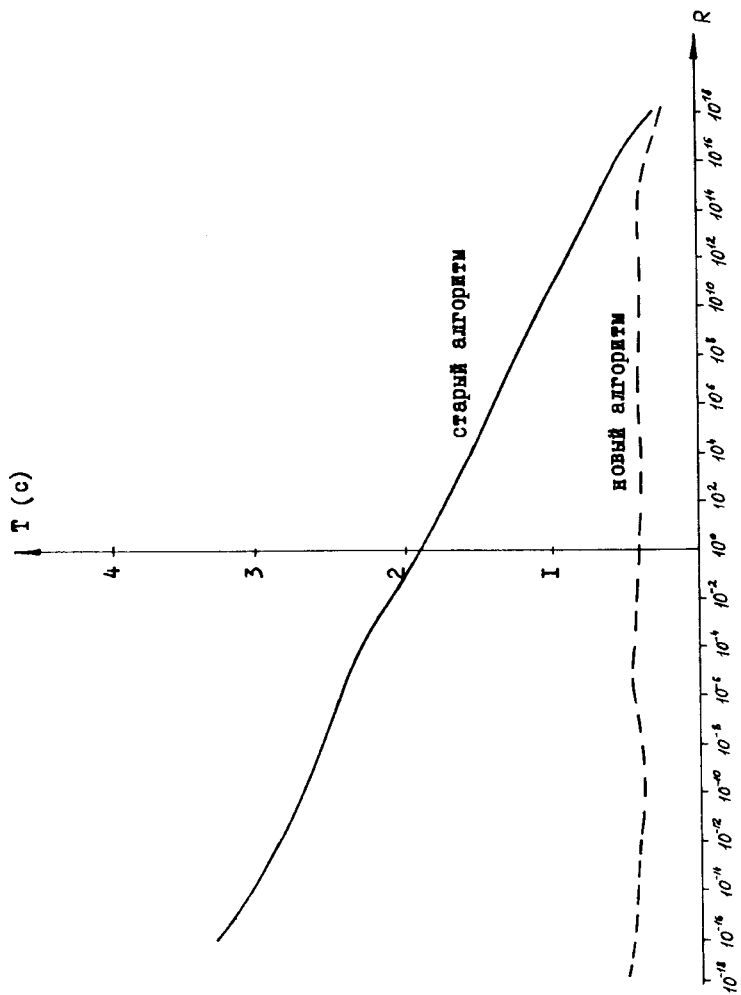
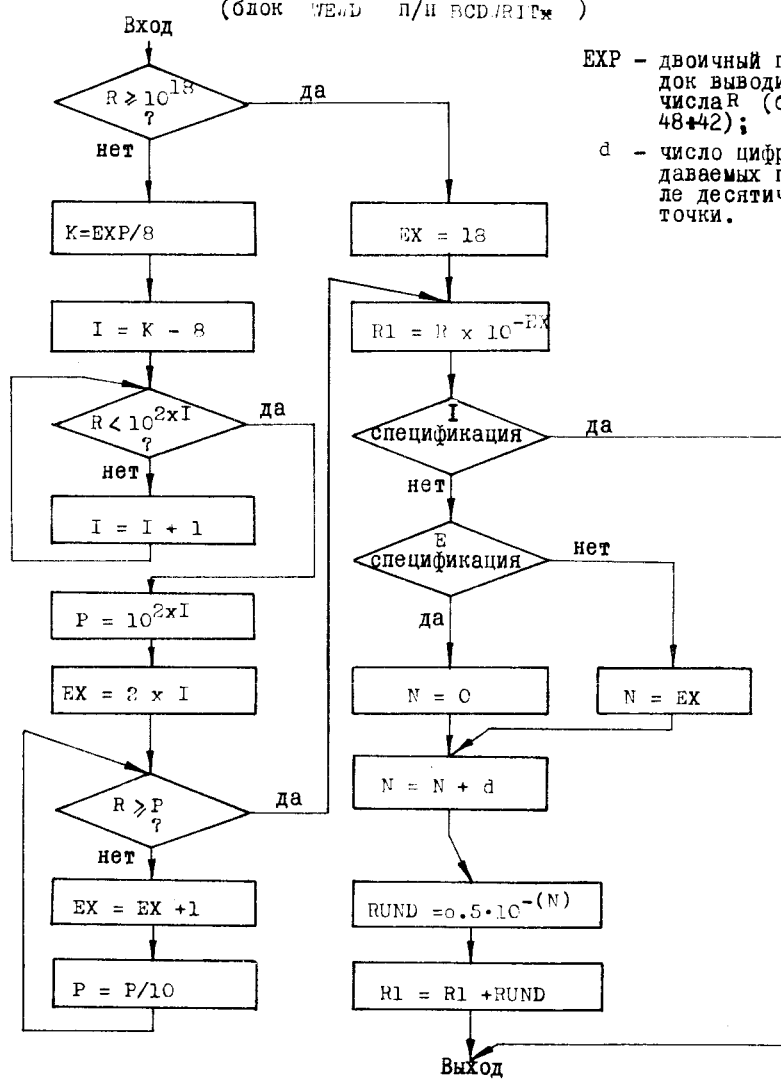


Рис. 1в.

Принципиальная блок-схема вычисления EX и подготовки к декодированию

(блок WEXP п/и WCDARIT*)



EXP - двоичный порядок выводимого числа R (биты 48+42);

d - число цифр выдаваемых после десятичной точки.

Рис. 1с.

НEXТЛЕТ* Блок-схема декодирования чисел (блоки НEXТЛЕТ* и РК* подпрограммы ВСДWRIT*).

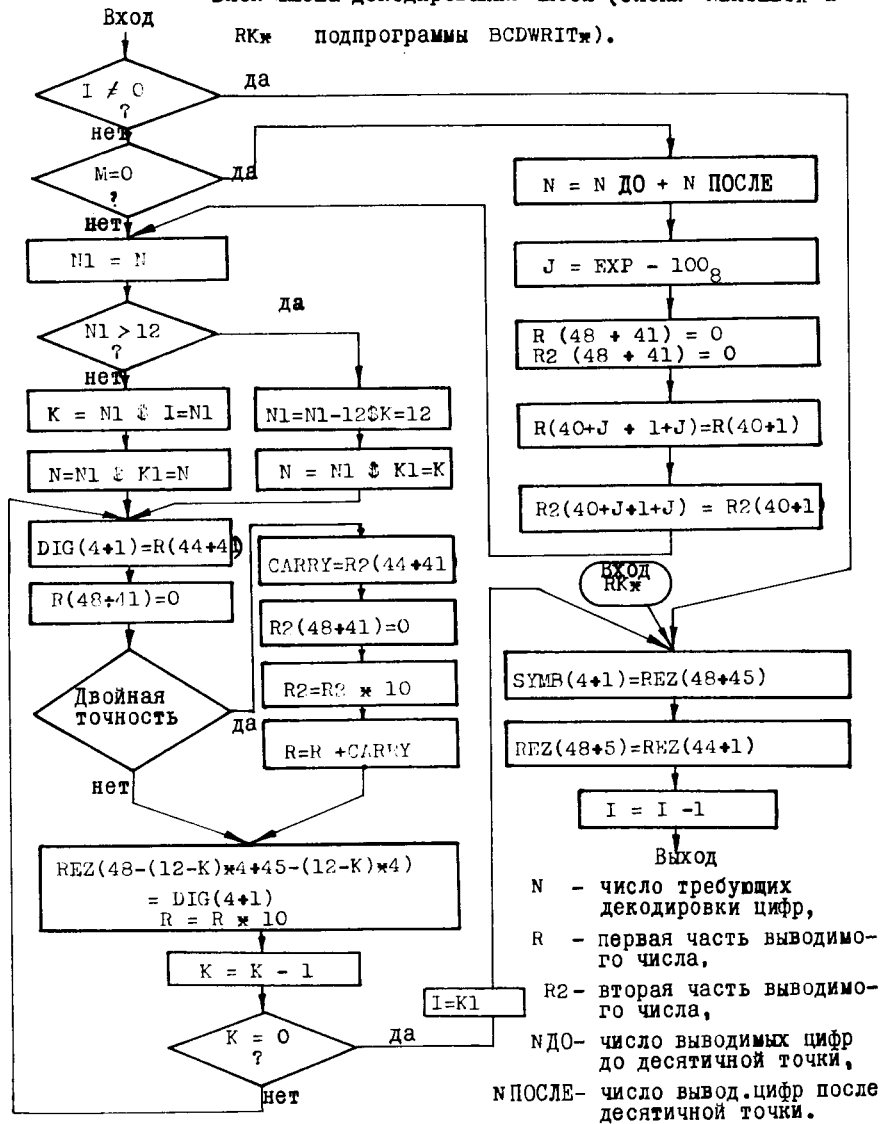


Рис. 1д.

Таблица 1е.

Порядок числа	Спецификация Е 8.1		Спецификация Е 13.5		Спецификация Е 15.8		Спецификация Е 19.11	
	старый алгоритм	новый алгоритм	старый алг.	новый алг.	старый алг.	новый алг.	старый алг.	новый алг.
10 ¹⁸	9.61	7.74	10.70	9.11	11.08	9.79	13.71	11.05
10 ¹⁵	10.21	8.17	10.93	9.57	13.08	10.13	14.07	11.40
10 ¹²	10.47	8.29	12.98	9.61	13.33	10.17	14.23	11.45
10 ⁹	10.72	8.13	12.24	9.42	12.61	10.07	16.15	11.39
10 ⁶	11.78	7.95	12.70	9.46	14.66	10.17	15.91	11.45
10 ³	12.45	7.97	14.67	9.38	15.01	10.04	16.35	11.35
10 ⁰	14.32	8.05	15.14	9.43	15.53	10.14	18.47	11.40
10 ⁻³	15.27	7.87	15.86	9.36	17.89	10.08	19.47	11.33
10 ⁻⁶	15.45	7.96	18.04	9.40	18.43	10.09	20.73	11.38
10 ⁻⁹	16.39	7.81	18.63	9.32	19.00	10.27	21.75	11.29
10 ⁻¹²	18.26	7.94	19.52	9.60	19.91	11.60	25.67	11.35
10 ⁻¹⁵	19.41	7.95	20.14	9.30	22.22	11.50	23.33	11.28
10 ⁻¹⁸	20.33	8.24	22.82	9.62	23.13	11.47	24.63	11.74

Сравнительные оценки времени, затрачиваемого на подготовку чисел к декодированию и декодированию по старому и новому алгоритмам (на 5000 чисел) в секундах.

ЛИТЕРАТУРА

1. "Язык ФОРТРАН" (Под редакцией В.П.Ширикова). II-4818, Дубна, 1969.
2. Г.Л.Мазный. "Мониторная система "ДУБНА" (Руководство для пользователей. Под редакцией В.Ю.Веретеннова, И.Н.Силина, В.П.Ширикова). II-5974, Дубна, 1971.
3. Fortran Extended Reference Manual Publ. No 60329100, Control Data Corporation, USA, 1971.
4. BESM INFORMATIONEN . Издание вычислительного центра Института физики высоких энергий Академии Наук ГДР, VI-B 70-01, Берлин-Цойтен, 1970.
- ✓ 5. Ю.П.Залаторюс. Новый вариант ввода-вывода в системе "ДУБНА". (Подробное описание). BI-IO-8771, Дубна, 1975.
6. В. А. Загинайко и др. Реализация обработки операторов обмена с внешними устройствами языка ФОРТРАН на машине БЭСМ-6. Б2-II-3507, Дубна, 1968.

Рукопись поступила в издательский отдел
13 мая 1975 г.